

Contents lists available at [SciVerse ScienceDirect](http://SciVerse.ScienceDirect.com)

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcsMore on bisimulations for higher order π -calculus

Zining Cao*

College of Computer Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China
 Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, China

ARTICLE INFO

Article history:

Received 28 July 2008

Received in revised form 3 November 2011

Accepted 13 April 2012

Communicated by M. Nielsen

Keywords:

Process calculi

Bisimulations

Higher order π -calculus

ABSTRACT

In this paper, we prove the coincidence between strong/weak context bisimulation and strong/weak normal bisimulation for higher order π -calculus, which generalizes Sangiorgi's work. To achieve this aim, we introduce indexed higher order π -calculus, which is similar to higher order π -calculus except that every prefix of any process is assigned indices. Furthermore we present corresponding indexed bisimulations for this calculus, and prove the equivalence between these indexed bisimulations. Based on this result, we prove the main result of this paper, i.e., the equivalence between strong/weak context bisimulation and strong/weak normal bisimulation.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Since Milner, Parrow and Walker's paper in 1992 [14], π -calculus has received ever growing interest, mainly for two reasons: it is well suited for specifying properties of concurrent systems and its mathematical theory is elegant. In Sangiorgi's dissertation [16], some interesting bisimulations for higher order π -calculus were presented, such as barbed equivalence, context bisimulation and normal bisimulation. Barbed equivalence can be regarded as a uniform definition of bisimulation for a variety of concurrency calculi. Context bisimulation is a very intuitive definition of bisimulation for higher order π -calculus, but it is heavy to handle, due to the appearance of universal quantifications in its definition. In the definition of normal bisimulation, all universal quantifications disappeared, therefore normal bisimulation is a very economic characterization of bisimulation for higher order π -calculus.

The main difficulty with definitions of context bisimulation and barbed equivalence that involve quantification over contexts is that they are often awkward to work with directly. It is therefore important to look for more tractable characterizations of the bisimulations. In [16], the equivalence among weak normal bisimulation, weak context bisimulation and weak barbed equivalence was proved, but the proof method cannot be adopted to prove the equivalence between strong context and normal bisimulations. Sangiorgi presented it as an open problem in [16]. The essential reason is that the proof relies on some central technical results, like the factorization theorem and the full abstraction of the mapping to triggered agents, which do not hold in the case of strong context bisimulation.

Up until now, the only results we are aware of on the equivalence between context and normal bisimulations are in [16,17,4,10]. All these papers only treated the case of weak bisimulations. In process calculi, almost all results about weak bisimulations can be easily generalized to the case of strong bisimulations. Roughly speaking, tau action is abstracted in weak bisimulations but not in strong bisimulations. In general, we have more interest for weak bisimulations than strong bisimulations since we usually do not care the internal computation in a computer system. But strong bisimulation is still an important concept. In [13], strong bisimulation was presented as a basic equivalence for process calculi. For example,

* Correspondence to: College of Computer Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China.
 Tel.: +86 25 84896490 12322.

E-mail address: caozn@nuaa.edu.cn.

if we want to match both the interface property and the internal computation steps of two systems, strong bisimulation is a suitable criterion. As far as we know, no paper gives the proof of equivalence between strong context and normal bisimulations. In [16,17], this problem was stated as an open problem. The main difficulty is that the strategy of proof for the equivalence between weak context and normal bisimulations does not work for the case of strong bisimulations. For the case of weak bisimulations, the translation to trigger processes will bring some redundant tau actions. Since weak bisimulations abstract from tau action, the problem is inessential. For the case of strong bisimulations, the situation is different. We have to match these redundant tau actions to prove that two processes are bisimilar. Therefore we need some new strategies of proof to solve the problem. The proof strategies may be also useful for other problems. For example, the equivalence between weak context and normal bisimulations can be also proved by this methods.

The main aim of this paper is to give a uniform proof for the equivalence of normal and context bisimulations in both strong case and weak case. Especially, we give a proof of the coincidence between strong normal bisimulation and strong context bisimulation, which solves an open problem presented by Sangiorgi in [16,17]. To achieve this aim, we introduce indexed processes and propose some bisimulations on indexed processes. Since it is straightforward to show that strong context bisimulation implies strong normal bisimulation, to get the proposition, we only need to prove that two processes are strongly context bisimilar if they are strongly normal bisimilar. We prove this claim by introducing the concept of indexed processes and corresponding bisimulations on indexed processes. The key point is that since indices distinguish tau actions, we can match the redundant tau actions assigned by the same special indices. Furthermore, we also prove the coincidence between indexed context bisimulation and indexed reduction congruences.

This paper is organized as follows: Section 2 gives a brief review of syntax and operational semantics of the higher order π -calculus, then recalls the definitions of context and normal bisimulations. Section 3 introduces indexed higher order π -calculus and some indexed bisimulations. The equivalence between these indexed bisimulations is also proved. In Section 4 we give a proof for the equivalence between strong/weak context bisimulation and strong/weak normal bisimulation. The paper is concluded in Section 5.

2. Higher order π -calculus

2.1. Syntax and labelled transition system of higher order π -calculus

In this section we briefly recall the syntax and labelled transition system of the higher order π -calculus. Similar to [17], we only focus on a second-order fragment of the higher order π -calculus, i.e., there is no abstraction in this fragment.

We assume a set N of names, ranged over by a, b, c, \dots and a set Var of process variables, ranged over by X, Y, Z, U, \dots . We use E, F, P, Q, \dots to stand for processes. Pr denotes the set of all processes.

We first give the grammar for the higher order π -calculus processes as follows:

$$P ::= 0 \mid U \mid \pi.P \mid P_1|P_2 \mid (\nu a)P \mid !P$$

π is called a prefix and can have one of the following forms:

$$\pi ::= \tau \mid l \mid \bar{l} \mid a(U) \mid \bar{a}\langle P \rangle$$

where τ is a tau prefix; l is a first order input prefix; \bar{l} is a first order output prefix; $a(U)$ is a higher order input prefix and $\bar{a}\langle P \rangle$ is a higher order output prefix.

For higher order π -calculus, in each process of the form $(\nu a)P$, P is the scope of binder (νa) . An occurrence of a in P is said to be free iff it does not lie within the scope of a binding occurrence of a . The set of names occurring free in P is denoted as $fn(P)$. An occurrence of a name in P is said to be bound if it is not free, we write the set of bound names as $bn(P)$. $n(P)$ denotes the set of names of P , i.e., $n(P) = fn(P) \cup bn(P)$. We use $n(P, Q)$ to denote $n(P) \cup n(Q)$. Higher order input prefix $a(U).P$ binds all free occurrences of U in P . The set of variables occurring free in P is denoted as $fv(P)$. We write the set of bound variables in P as $bv(P)$. $v(P)$ denotes the set of variables of P , i.e., $v(P) = fv(P) \cup bv(P)$. A process is closed if it has no free variable; it is open if it may have free variables. Pr^c is the set of all closed processes. Processes P and Q are α -convertible, $P \equiv_\alpha Q$, if Q can be obtained from P by a finite number of changes of bound names and bound variables.

The set of all closed processes, i.e., the processes which have no free variable, is denoted as Pr^c .

The structural congruence relation is the smallest congruence generated by the following laws:

$$\begin{aligned} P|0 &\equiv P, & P_1|P_2 &\equiv P_2|P_1, & P_1|(P_2|P_3) &\equiv (P_1|P_2)|P_3, & (\nu a)0 &\equiv 0, \\ (\nu a)(\nu b)P &\equiv (\nu b)(\nu a)P, & P|(\nu a)Q &\equiv (\nu a)(P|Q) & \text{if } a \notin fn(P), \\ a(U).P &\equiv a(V).(P\{V/U\}) & \text{if } V \notin v(P). \end{aligned}$$

The operational semantics of higher order processes is given in Table 1. We have omitted the symmetric cases for the PAR, COM1 and COM2 rules.

2.2. Bisimulations in higher order π -calculus

Context and normal bisimulations were presented in [16,17] to describe the behavioural equivalences for higher order π -calculus. In the following, we abbreviate $P\{E/U\}$ as $P\langle E \rangle$.

Table 1
Operational semantics of higher order π -calculus.

$$\begin{array}{l}
 \text{ALP} : \frac{P \xrightarrow{\alpha} P'}{Q \xrightarrow{\alpha} Q'} P \equiv Q, P' \equiv Q' \quad \text{TAU} : \tau.P \xrightarrow{\tau} P \\
 \text{OUT1} : \bar{l}.P \xrightarrow{\bar{l}} P \quad \text{IN1} : l.P \xrightarrow{l} P \\
 \text{OUT2} : \bar{a}(E).P \xrightarrow{\bar{a}(E)} P \quad \text{IN2} : a(U).P \xrightarrow{a(E)} P\{E/U\} \\
 \text{PAR} : \frac{P \xrightarrow{\alpha} P' \quad Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P'|Q} \text{bn}(\alpha) \cap \text{fn}(Q) = \emptyset \\
 \text{COM1} : \frac{P \xrightarrow{\bar{l}} P' \quad Q \xrightarrow{l} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \frac{(\nu \tilde{b})\bar{a}(E)}{P|Q \xrightarrow{\tau} P'|Q'} \frac{a(E)}{Q' \xrightarrow{a(E)} Q'} \tilde{b} \cap \text{fn}(Q) = \emptyset \\
 \text{COM2} : \frac{P \xrightarrow{\bar{l}} P' \quad Q \xrightarrow{l} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \frac{(\nu \tilde{b})\bar{a}(E)}{P|Q \xrightarrow{\tau} P'|Q'} \frac{a(E)}{Q' \xrightarrow{a(E)} Q'} \tilde{b} \cap \text{fn}(Q) = \emptyset \\
 \text{RES} : \frac{P \xrightarrow{\alpha} P' \quad a \notin \text{fn}(\alpha)}{(\nu a)P \xrightarrow{\alpha} (\nu a)P'} \quad \text{REP} : \frac{P \xrightarrow{\alpha} P' \quad !P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P'} \\
 \text{OPEN} : \frac{P \xrightarrow{\bar{a}(E)} P' \quad a \neq b, b \in \text{fn}(E) - \tilde{c}}{(\nu b, \tilde{c})P \xrightarrow{\bar{a}(E)} P'}
 \end{array}$$

Definition 1. A symmetric relation $R \subseteq Pr^c \times Pr^c$ is a strong context bisimulation if $P R Q$ implies:

- (1) whenever $P \xrightarrow{\alpha} P'$, there exists Q' such that $Q \xrightarrow{\alpha} Q'$, where α is in the form of τ or l or \bar{l} , and $P' R Q'$;
- (2) whenever $P \xrightarrow{a(E)} P'$, there exists Q' such that $Q \xrightarrow{a(E)} Q'$ and $P' R Q'$;
- (3) whenever $P \xrightarrow{(\nu \tilde{b})\bar{a}(E)} P'$, there exist Q', F, \tilde{c} such that $Q \xrightarrow{(\nu \tilde{c})\bar{a}(F)} Q'$ and for all $C(U)$ with $\text{fn}(C(U)) \cap \{\tilde{b}, \tilde{c}\} = \emptyset$, $(\nu \tilde{b})(P'|C(E)) R (\nu \tilde{c})(Q'|C(F))$, where $C(U)$ is a process containing a unique free variable U .

We write $P \sim_{ct} Q$ if P and Q are strongly context bisimilar.

Distinguished from strong context bisimulation, strong normal bisimulation does not have universal quantifications in the clauses of its definition. In the following, a name is called fresh in a statement if it is different from any other name occurring in the processes of the statement.

Definition 2. A symmetric relation $R \subseteq Pr^c \times Pr^c$ is a strong normal bisimulation if $P R Q$ implies:

- (1) whenever $P \xrightarrow{\alpha} P'$, there exists Q' such that $Q \xrightarrow{\alpha} Q'$, where α is in the form of τ or l or \bar{l} , and $P' R Q'$;
- (2) whenever $P \xrightarrow{a(\tilde{m}.0)} P'$, there exists Q' such that $Q \xrightarrow{a(\tilde{m}.0)} Q'$ and $P' R Q'$, where m is a fresh name, i.e., m is different from any name in P and Q ;
- (3) whenever $P \xrightarrow{(\nu \tilde{b})\bar{a}(E)} P'$, there exist Q', F, \tilde{c} such that $Q \xrightarrow{(\nu \tilde{c})\bar{a}(F)} Q'$ and $(\nu \tilde{b})(P'|!m.E) R (\nu \tilde{c})(Q'|!m.F)$, where m is a fresh name, i.e., m is different from any name in P, Q, E and F .

We write $P \sim_{Nr} Q$ if P and Q are strongly normal bisimilar.

In the following, we use $\xRightarrow{\varepsilon}$ to abbreviate the reflexive and transitive closure of $\xrightarrow{\tau}$, and use $\xRightarrow{\alpha}$ to abbreviate $\xRightarrow{\varepsilon} \xrightarrow{\alpha} \xRightarrow{\varepsilon}$. By abstracting the tau action, we can get the following formal definitions of weak bisimulations:

Definition 3. A symmetric relation $R \subseteq Pr^c \times Pr^c$ is a weak context bisimulation if $P R Q$ implies:

- (1) whenever $P \xRightarrow{\varepsilon} P'$, there exists Q' such that $Q \xRightarrow{\varepsilon} Q'$ and $P' R Q'$;
- (2) whenever $P \xRightarrow{\alpha} P'$, there exists Q' such that $Q \xRightarrow{\alpha} Q'$, where α is in the form of l or \bar{l} , and $P' R Q'$;
- (3) whenever $P \xRightarrow{a(E)} P'$, there exists Q' such that $Q \xRightarrow{a(E)} Q'$ and $P' R Q'$;
- (4) whenever $P \xRightarrow{(\nu \tilde{b})\bar{a}(E)} P'$, there exist Q', F, \tilde{c} such that $Q \xRightarrow{(\nu \tilde{c})\bar{a}(F)} Q'$ and for all $C(U)$ with $\text{fn}(C(U)) \cap \{\tilde{b}, \tilde{c}\} = \emptyset$, $(\nu \tilde{b})(P'|C(E)) R (\nu \tilde{c})(Q'|C(F))$.

We write $P \approx_{ct} Q$ if P and Q are weakly context bisimilar.

Definition 4. A symmetric relation $R \subseteq Pr^c \times Pr^c$ is a weak normal bisimulation if $P R Q$ implies:

- (1) whenever $P \xRightarrow{\varepsilon} P'$, there exists Q' such that $Q \xRightarrow{\varepsilon} Q'$ and $P' R Q'$;
- (2) whenever $P \xRightarrow{\alpha} P'$, there exists Q' such that $Q \xRightarrow{\alpha} Q'$, where α is in the form of l or \bar{l} , and $P' R Q'$;
- (3) whenever $P \xRightarrow{a(\tilde{m}.0)} P'$, there exists Q' such that $Q \xRightarrow{a(\tilde{m}.0)} Q'$ and $P' R Q'$, where m is a fresh name;
- (4) whenever $P \xRightarrow{(\nu \tilde{b})\bar{a}(E)} P'$, there exist Q', F, \tilde{c} such that $Q \xRightarrow{(\nu \tilde{c})\bar{a}(F)} Q'$ and $(\nu \tilde{b})(P'|!m.E) R (\nu \tilde{c})(Q'|!m.F)$, where m is a fresh name.

We write $P \approx_{Nr} Q$ if P and Q are weakly normal bisimilar.

3. Indexed processes and indexed bisimulations

In [16,17], the equivalence between weak context bisimulation and weak normal bisimulation was proved. In the proof, the factorization theorem was firstly given. It allows us to factorize out certain subprocesses of a given process. Thus, a complex process can be decomposed into the parallel composition of simpler processes. Then the concept of triggered processes was introduced, which is the key step in the proof. Triggered processes represent a sort of normal form for the processes. Most importantly, there is a very simple characterization of context bisimulation on triggered processes, called triggered bisimulation. By the factorization theorem, a process can be transformed to a triggered process. The transform allows us to use the simpler theory of triggered processes to reason about the set of all processes. In [16,17], weak context bisimulation was firstly proved to be equivalent to weak triggered bisimulation on triggered processes, then by the mapping from general processes to triggered processes, the equivalence between weak context bisimulation and weak normal bisimulation was proved.

In the case of strong bisimulations, the above proof strategy does not work. The main problem is that the mapping to triggered processes brings some redundant tau actions. Since weak bisimulations abstract from tau action, the full abstraction of the mapping to triggered processes holds. But in the case of strong bisimulations, the triggered mapping does not preserve the strong bisimulations, and therefore some central technical results in [16,17], like the factorization theorem, are not true in the strong case.

To resolve this difficulty we introduced the concept of indexed processes and the indexed version of context and normal bisimulations. Roughly, the actions of indexed processes have added indices, which are used to identify in which component or between which components an action takes place. Indexed bisimulations with respect to an index set S abstract the indexed tau action of the form $\{\tau\}_{i,i}$ for any $i \in S$, but distinguish the indexed tau action of the form $\{\tau\}_{i,j}$ if $i \notin S$ or $j \notin S$ or $i \neq j$. To study the relation between indexed bisimulations, we present a translation Tr^n which transforms every indexed process M into the indexed triggered process $Tr^n[M]$ with respect to index n . One can see that the mapping from M to $Tr^n[M]$ brings redundant indexed tau actions $\{\tau\}_{n,n}$. Therefore indexed triggered mapping preserves the indexed bisimulations with respect to $S \cup \{n\}$ for any S . Similarly, we also have the indexed version of the factorization theorem. Following the proof strategy in [16,17], we prove the equivalence between indexed context bisimulation and indexed normal bisimulation.

Furthermore, when S is the empty set \emptyset , we discuss the relation between indexed bisimulations and strong bisimulations, and get the proposition: $P \sim_{Nr} Q \Leftrightarrow \{P\}_k \simeq_{Nr}^{\emptyset} \{Q\}_k \Leftrightarrow Tr^n[\{P\}_k] \simeq_{Tr}^{\{n\}} Tr^n[\{Q\}_k] \Leftrightarrow \{P\}_k \simeq_{Ct}^{\emptyset} \{Q\}_k \Leftrightarrow P \sim_{Ct} Q$, where $k \neq n$. This solves the open problem in [16,17]. We also apply the proof idea to the case of weak bisimulations. When S is the full index set I , we study the relation between indexed bisimulations and weak bisimulations, and get the proposition: $P \approx_{Nr} Q \Leftrightarrow \{P\}_k \simeq_{Nr}^I \{Q\}_k \Leftrightarrow Tr^n[\{P\}_k] \simeq_{Tr}^I Tr^n[\{Q\}_k] \Leftrightarrow \{P\}_k \simeq_{Ct}^I \{Q\}_k \Leftrightarrow P \approx_{Ct} Q$, where I is the full index set. Therefore the proof presented here seems to be a uniform approach to the equivalence between strong/weak context bisimulation and strong/weak normal bisimulation.

3.1. Syntax and labelled transition system of indexed higher order π -calculus

The aim of this paper is to propose a general technique for showing the correspondence of context and normal bisimulations in both the strong and weak cases, by relying on a notion of indexed processes. Roughly, the intention is that indexed processes allow the labelled transition system semantics to record in action labels the indices of the interacting components. This mechanism is then used to filter out some tau transitions in the considered definition of bisimulation.

Now we introduce the concept of indexed processes. The index set I , w.l.o.g., will be the set of natural numbers. Intuitively, the concept of index can be viewed as the name or location of components. The class of the indexed processes IPr is built similar to Pr , except that every prefix is assigned indices. We usually use K, L, M, N to denote indexed processes.

The formal definition of an indexed process is given as follows:

$$M ::= 0 \mid U \mid I\pi.M \mid M_1|M_2 \mid (\nu a)M \mid !M.$$

$I\pi$ is called an indexed prefix and can be an indexed tau prefix or an indexed input prefix or an indexed output prefix:

$$I\pi ::= \{\tau\}_{i,j} \mid \{l\}_i \mid \{\bar{a}(U)\}_i \mid \{\bar{a}(N)\}_i$$

where $i, j \in \text{index set } I$, N is an indexed process.

The notations of free name, bound name, free variable, bound variable etc. can be given similar to the original higher order π -calculus. IPr^c is the set of all closed indexed processes. Indexed processes M and N are α -convertible, $M \equiv_\alpha N$, if N can be obtained from M by a finite number of changes of bound names and bound variables.

Definition 5. The set of all indices that occur in M , $ix(M)$, is defined inductively as follows:

- (1) if $M = 0$ or U , then $ix(M) ::= \emptyset$;
- (2) if $M = I\pi.M_1$, then $ix(M) ::= ix(I\pi) \cup ix(M_1)$, where $ix(I\pi) ::= \{i, j\}$ if $I\pi$ is in the form of $\{\tau\}_{i,j}$; $ix(I\pi) ::= \{i\} \cup ix(N)$ if $I\pi$ is in the form of $\{\bar{a}(N)\}_i$; $ix(I\pi) ::= \{i\}$ if $I\pi$ is in the form of $\{l\}_i$ or $\{\bar{a}(U)\}_i$.
- (3) if $M = M_1|M_2$, then $ix(M) ::= ix(M_1) \cup ix(M_2)$;
- (4) if $M = (\nu a)M_1$, then $ix(M) ::= ix(M_1)$;
- (5) if $M = !M_1$, then $ix(M) ::= ix(M_1)$.

We use $ix(M, N)$ to denote $ix(M) \cup ix(N)$.

Table 2Operational semantics of indexed higher order π -calculus.

$ALP : \frac{M \xrightarrow{I\alpha} M'}{N \xrightarrow{I\alpha} N'} M \equiv N, M' \equiv N'$	$TAU : \{\tau\}_{i,j}.M \xrightarrow{\{\tau\}_{i,j}} M$
$OUT1 : \{\bar{l}\}_i.M \xrightarrow{\{\bar{l}\}_i} M$	$IN1 : \{l\}_i.M \xrightarrow{\{l\}_i} M$
$OUT2 : \{\bar{a}\langle K \rangle\}_i.M \xrightarrow{\{\bar{a}\langle K \rangle\}_i} M$	$IN2 : \{a\langle U \rangle\}_i.M \xrightarrow{\{a\langle K \rangle\}_i} M\{K/U\}$
$PAR : \frac{M \xrightarrow{I\alpha} M'}{M N \xrightarrow{I\alpha} M' N} bn(I\alpha) \cap fn(N) = \emptyset$	
$COM1 : \frac{\frac{M \xrightarrow{\{l\}_i} M'}{M N \xrightarrow{\{l\}_i} M' N} \frac{N \xrightarrow{\{l\}_j} N'}{M N \xrightarrow{\{\tau\}_{i,j}} (M' N')}$	
$COM2 : \frac{M N \xrightarrow{\{(\nu \tilde{b})\bar{a}\langle K \rangle\}_i} M'}{M N \xrightarrow{\{\tau\}_{i,j}} (M' N')} \frac{N \xrightarrow{\{a\langle K \rangle\}_j} N'}{M N \xrightarrow{\{\tau\}_{i,j}} (M' N')} \tilde{b} \cap fn(N) = \emptyset$	
$RES : \frac{M \xrightarrow{I\alpha} M'}{(va)M \xrightarrow{I\alpha} (va)M'} a \notin n(I\alpha)$	$REP : \frac{M M \xrightarrow{I\alpha} M'}{!M \xrightarrow{I\alpha} M'}$
$OPEN : \frac{M \xrightarrow{\{(\nu \tilde{b})\bar{a}\langle K \rangle\}_i} M'}{(vb)M \xrightarrow{\{(\nu \tilde{b}, \tilde{c})\bar{a}\langle K \rangle\}_i} M'} a \neq b, b \in fn(K) - \tilde{c}$	

In the remainder of this paper, $\{P\}_i$ is an abbreviation for the indexed process with the same given index i on every prefix in the scope of P . The formal definition is given below.

Definition 6. $\{P\}_i$ is defined inductively as follows:

- (1) $\{0\}_i ::= 0$;
- (2) $\{U\}_i ::= U$;
- (3) $\{\tau.P\}_i ::= \{\tau\}_{i,i}.\{P\}_i$;
- (4) $\{l.P\}_i ::= \{l\}_i.\{P\}_i$;
- (5) $\{\bar{l}.P\}_i ::= \{\bar{l}\}_i.\{P\}_i$;
- (6) $\{a(U).P\}_i ::= \{a\langle U \rangle\}_i.\{P\}_i$;
- (7) $\{\bar{a}(E).P\}_i ::= \{\bar{a}\langle \{E\}_i \rangle\}_i.\{P\}_i$;
- (8) $\{P_1|P_2\}_i ::= \{P_1\}_i|\{P_2\}_i$;
- (9) $\{(va)P\}_i ::= (va)\{P\}_i$;
- (10) $\{!P\}_i ::= !\{P\}_i$.

The structural congruence relation is the smallest congruence generated by the following laws:

$$\begin{aligned}
 M|0 &\equiv M, & M_1|M_2 &\equiv M_2|M_1, & M_1|(M_2|M_3) &\equiv (M_1|M_2)|M_3, \\
 (va)0 &\equiv 0, & (va)(vb)M &\equiv (vb)(va)M, & M|(va)N &\equiv (va)(M|N) \text{ if } a \notin fn(M), \\
 a(U).M &\equiv a(V).(M\{V/U\}) \text{ if } V \notin v(M).
 \end{aligned}$$

In the labelled transition system of indexed higher order π -calculus, the label on the transition arrow is an indexed action, whose definition is given as follows:

$$I\alpha ::= \{\tau\}_{i,j} \mid \{l\}_i \mid \{\bar{l}\}_i \mid \{a\langle K \rangle\}_i \mid \{\bar{a}\langle K \rangle\}_i \mid \{(\nu \tilde{b})\bar{a}\langle K \rangle\}_i$$

where $\{\tau\}_{i,j}$ is an indexed tau action, $\{l\}_i$ is an indexed first order input action, $\{\bar{l}\}_i$ is an indexed first order output action, $\{a\langle K \rangle\}_i$ is an indexed higher order input action, and $\{\bar{a}\langle K \rangle\}_i$ and $\{(\nu \tilde{b})\bar{a}\langle K \rangle\}_i$ are indexed higher order output actions.

We write $bn(I\alpha)$ to represent the set of names bound in $I\alpha$, which is $\{b\}$ if $I\alpha$ is $\{(\nu \tilde{b})\bar{a}\langle K \rangle\}_i$ and \emptyset otherwise. $n(I\alpha)$ is the set of names that occur in $I\alpha$.

The operational semantics of indexed processes is given in Table 2. Similar to Table 1, we have omitted the symmetric cases of the PAR, COM1 and COM2 rules. The main difference between Tables 1 and 2 is that the label $I\alpha$ on the transition arrow is in the form of $\{\alpha\}_i$ or $\{\tau\}_{i,j}$. If we adopt the distributed view, $\{\alpha\}_i$ can be regarded as an input or output action performed by component i , and $\{\tau\}_{i,j}$ can be regarded as a communication between components i and j .

3.2. Indexed context bisimulation and indexed normal bisimulation

Now we can give the concept of indexed context bisimulation and indexed normal bisimulation for indexed processes. In the remainder of this paper, we abbreviate $M\{K/U\}$ as $M\langle K \rangle$. In the following, we use $M \xrightarrow{\varepsilon, S} M'$ to abbreviate $M \xrightarrow{\{\tau\}_{i_1, i_1}} \dots \xrightarrow{\{\tau\}_{i_n, i_n}} M'$, where $i_1, \dots, i_n \in S$. Intuitively, $M \xrightarrow{\varepsilon, S} M'$ denotes the transition from M to M' through several internal communications from matching indices in S . Suppose $i_1 \neq i_2$, since $\{\tau\}_{i_1, i_2}$ is regarded as a communication between indices i_1 and i_2 , which cannot be abstracted, $\{\tau\}_{i_1, i_2}$ is not included in $M \xrightarrow{\{\tau\}_{i_1, i_1}}$

... $\xrightarrow{\{\tau\}_{i_n, i_n}} M'$, where $i_1, \dots, i_n \in S \subseteq I$. We use $M \xrightarrow{I\alpha, S} M'$ to abbreviate $M \xrightarrow{\varepsilon, S} \xrightarrow{I\alpha} \xrightarrow{\varepsilon, S} M'$. An index is called fresh in a statement if it is different from any other index occurring in the processes of the statement. Let us see two examples. For the transition $(\nu a)((\nu b)(\{\bar{a}\}_i.0|\{\bar{b}\}_j.0|\{a\}_i.\{b\}_j.0)) \xrightarrow{\{\tau\}_{i,i} \{\tau\}_{j,j}} 0$, we can abbreviate it as $(\nu a)((\nu b)(\{\bar{a}\}_i.0|\{\bar{b}\}_j.0|\{a\}_i.\{b\}_j.0)) \xrightarrow{\varepsilon, \{i,j\}} 0$. Similarly, since $(\nu a)((\nu b)(\{\bar{a}\}_i.0|\{\bar{b}\}_j.0|\{a\}_i.\{c\}_k.\{b\}_i.0)) \xrightarrow{\{\tau\}_{i,i} \{\tau\}_{j,j} \{\tau\}_{k,k}} 0$, we can abbreviate it as $(\nu a)((\nu b)(\{\bar{a}\}_i.0|\{\bar{b}\}_j.0|\{a\}_i.\{c\}_k.\{b\}_i.0)) \xrightarrow{\{c\}_k, \{i,j\}} 0$.

This paper's main result states that strong context bisimulation coincides with strong normal bisimulation. Technically, the proof rests on the notion of indexed bisimulations. The idea is to generalize the usual notion of weak bisimulations so that tau actions can be abstracted selectively, depending on a chosen set of indices S . The cases $S = \emptyset$ and $S = I$ correspond to strong and weak bisimulations respectively.

Definition 7. Let M, N be two closed indexed processes, and $S \subseteq I$ be an index set, we write $M \simeq_{\text{ct}}^S N$, if there is a symmetric relation R and $M R N$ implies:

- (1) whenever $M \xrightarrow{\varepsilon, S} M'$, there exists N' such that $N \xrightarrow{\varepsilon, S} N'$ and $M' R N'$;
- (2) whenever $M \xrightarrow{I\alpha, S} M'$, there exists N' such that $N \xrightarrow{I\alpha, S} N'$ and $M' R N'$, where $I\alpha$ is in the form of $\{\tau\}_{i,j}$ or $\{l\}_i$ or $\{\bar{l}\}_i$, $I\alpha \neq \{\tau\}_{i,i}$ for any $i \in S$;
- (3) whenever $M \xrightarrow{\{a(K)\}_{i,S}} M'$, there exists N' such that $N \xrightarrow{\{a(K)\}_{i,S}} N'$ and $M' R N'$;
- (4) whenever $M \xrightarrow{\{(\nu \bar{b})\bar{a}(K)\}_{i,S}} M'$, there exists N' such that $N \xrightarrow{\{(\nu \bar{c})\bar{a}(L)\}_{i,S}} N'$ and for any indexed process $C(U)$ with $\text{fn}(C(U)) \cap \{\bar{b}, \bar{c}\} = \emptyset$, $(\nu \bar{b})(M'|C(K)) R (\nu \bar{c})(N'|C(L))$.

We say that M and N are indexed context bisimilar with respect to S if $M \simeq_{\text{ct}}^S N$.

Indexed context bisimulation \simeq_{ct}^S can be extended to open indexed processes.

Definition 8. For two open indexed processes M, N , we write $M \simeq_{\text{ct}}^S N$ if $M\{\tilde{K}/\tilde{X}\} \simeq_{\text{ct}}^S N\{\tilde{K}/\tilde{X}\}$ for any \tilde{K} , where $\tilde{X} = \text{fv}(M) \cup \text{fv}(N)$.

Definition 9. Let M, N be two closed indexed processes, and $S \subseteq I$ be an index set, we write $M \simeq_{\text{nr}}^S N$, if there is a symmetric relation R and $M R N$ implies:

- (1) whenever $M \xrightarrow{\varepsilon, S} M'$, there exists N' such that $N \xrightarrow{\varepsilon, S} N'$ and $M' R N'$;
- (2) whenever $M \xrightarrow{I\alpha, S} M'$, there exists N' such that $N \xrightarrow{I\alpha, S} N'$ and $M' R N'$, where $I\alpha$ is in the form of $\{\tau\}_{i,j}$ or $\{l\}_i$ or $\{\bar{l}\}_i$, $I\alpha \neq \{\tau\}_{i,i}$ for any $i \in S$;
- (3) whenever $M \xrightarrow{\{a(\{\bar{m}\})_n.0\}_{i,S}} M'$, where m is a fresh name and n is a fresh index, there exists N' such that $N \xrightarrow{\{a(\{\bar{m}\})_n.0\}_{i,S}} N'$ and $M' R N'$;
- (4) whenever $M \xrightarrow{\{(\nu \bar{b})\bar{a}(K)\}_{i,S}} M'$, there exists N' such that $N \xrightarrow{\{(\nu \bar{c})\bar{a}(L)\}_{i,S}} N'$, and $(\nu \bar{b})(M'!\{m\}_n.K) R (\nu \bar{c})(N'!\{m\}_n.L)$ with a fresh name m and a fresh index n .

We say that M and N are indexed normal bisimilar with respect to S if $M \simeq_{\text{nr}}^S N$.

The above definitions have some geometric intuition. From a distributed view, $\{\tau\}_{i,i}$ is an internal communication in component i , and $\{\tau\}_{i,j}$, where $i \neq j$, represents an external communication between different components i and j . Therefore in Definitions 7 and 9, we regard $\{\tau\}_{i,i}$ as a private event in component i , which can be abstracted if i is in S , a chosen set of indices; and we view $\{\tau\}_{i,j}$ as a visible event between different components i and j .

For example, by the above definition, we have $(\nu a)(\{\bar{a}\}_n.0|\{a\}_n.M) \simeq_{\text{ct}}^{\{n\}} M$, $(\nu a)(\{\bar{a}\}_n.0|\{a\}_n.M) \not\simeq_{\text{ct}}^{\emptyset} M$ and $(\nu a)(\{\bar{a}\}_n.0|\{a\}_n.M) \simeq_{\text{nr}}^I M$.

3.3. Indexed triggered processes and indexed triggered bisimulation

The concept of triggered processes was introduced in [16,17]. The distinguishing feature of triggered processes is that every communication among them is the exchange of a trigger, where a trigger is an elementary process whose only functionality is to activate a copy of another process. In this section, we introduce the indexed version of triggered processes. An indexed triggered process can be seen as a sort of normal form for the indexed processes, and every communication among them is the exchange of an indexed trigger. We shall use indexed triggers to perform indexed process transformations which make the treatment of the constructs of indexed higher order processes easier.

The formal definition of indexed triggered process is given as follows:

$$M ::= 0 \mid U \mid \{\tau\}_{i,j}.M \mid \{l\}_i.M \mid \{\bar{l}\}_i.M \mid \{a(U)\}_i.M \mid (\nu m)(\{\bar{a}(\{\bar{m}\})_n.0\}_{i,S}.M \mid \{m\}_n.N) \text{ with } m \notin \text{fn}(M, N) \cup \{a\} \mid M_1 M_2 \mid (\nu a)M \mid !M.$$

The class of the indexed triggered processes is denoted as $ITPr$. The class of the closed indexed triggered processes is denoted as $ITPr^c$.

Definition 10. We give a mapping Tr^n which transforms every indexed process M into the indexed triggered process $Tr^n[M]$ with respect to index n . The mapping is defined inductively on the structure of M .

- (1) $Tr^n[0] ::= 0$;
- (2) $Tr^n[U] ::= U$;
- (3) $Tr^n[\{\tau\}_{i,j}.M] ::= \{\tau\}_{i,j}.Tr^n[M]$;
- (4) $Tr^n[\{l\}_i.M] ::= \{l\}_i.Tr^n[M]$;
- (5) $Tr^n[\{\bar{l}\}_i.M] ::= \{\bar{l}\}_i.Tr^n[M]$;
- (6) $Tr^n[\{a(U)\}_i.M] ::= \{a(U)\}_i.Tr^n[M]$;
- (7) $Tr^n[\{\bar{a}\langle N \rangle\}_i.M] ::= (vm)(\{\bar{a}\langle \bar{m} \rangle_n.0\})_i.Tr^n[M]! \{m\}_n.Tr^n[N]$, where m is a fresh name;
- (8) $Tr^n[M_1|M_2] ::= Tr^n[M_1]|Tr^n[M_2]$;
- (9) $Tr^n[(va)M] ::= (va)Tr^n[M]$;
- (10) $Tr^n[!M] ::= !Tr^n[M]$.

Transformation $Tr^n[\]$ may expand the number of $\{\tau\}_{n,n}$ steps in a process. But the behaviour is otherwise the same. The expansion is due to the fact that if in M a process N is transmitted and used k times then, in $Tr^n[M]$ k additional $\{\tau\}_{n,n}$ interactions are required to activate the copies of N .

For example, let $M \stackrel{def}{=} \{\bar{a}\langle N \rangle\}_i.L\{a(U)\}_j.(U|U)$, then $M \xrightarrow{\{\tau\}_{i,j}} L|N|N \stackrel{def}{=} M'$. In $Tr^n[M]$, this is simulated using two additional $\{\tau\}_{n,n}$ interactions:

$$\begin{aligned} Tr^n[M] &= (vm)(\{\bar{a}\langle \bar{m} \rangle_n.0\})_i.Tr^n[L]! \{m\}_n.Tr^n[N]|\{a(U)\}_j.(U|U) \\ &\xrightarrow{\{\tau\}_{i,j}} (vm)(Tr^n[L]! \{m\}_n.Tr^n[N]|\{\bar{m}\}_n.0|\{\bar{m}\}_n.0) \\ &\xrightarrow{\{\tau\}_{n,n} \ \{\tau\}_{n,n}} (vm)(Tr^n[L]|Tr^n[N]|Tr^n[N]! \{m\}_n.Tr^n[N]) \\ &\stackrel{\simeq_{Ct}^\emptyset}{\sim} Tr^n[L]|Tr^n[N]|Tr^n[N] \text{ since } m \text{ is a fresh name} \\ &= Tr^n[M']. \end{aligned}$$

Now we can give the indexed version of triggered bisimulation as follows.

Definition 11. Let M, N be two closed indexed triggered processes, and $S \subseteq I$ be an index set, we write $M \simeq_{Tr}^S N$, if there is a symmetric relation R and $M R N$ implies:

- (1) whenever $M \xrightarrow{\varepsilon, S} M'$, there exists N' such that $N \xrightarrow{\varepsilon, S} N'$ and $M' R N'$;
- (2) whenever $M \xrightarrow{I\alpha, S} M'$, there exists N' such that $N \xrightarrow{I\alpha, S} N'$ and $M' R N'$, where $I\alpha$ is in the form of $\{\tau\}_{i,j}$ or $\{l\}_i$ or $\{\bar{l}\}_i$, $I\alpha \neq \{\tau\}_{i,i}$ for any $i \in S$;
- (3) whenever $M \xrightarrow{\{a\langle \bar{m} \rangle_n.0\}_{i,S}} M'$, where m is a fresh name and n is a fresh index, there exists N' such that $N \xrightarrow{\{a\langle \bar{m} \rangle_n.0\}_{i,S}} N'$ and $M' R N'$;
- (4) whenever $M \xrightarrow{\{(vm)\bar{a}\langle \bar{m} \rangle_n.0\}_{i,S}} M'$, there exists N' such that $N \xrightarrow{\{(vm)\bar{a}\langle \bar{m} \rangle_n.0\}_{i,S}} N'$ and $M' R N'$.

We say that M and N are indexed triggered bisimilar with respect to S if $M \simeq_{Tr}^S N$.

Indexed triggered bisimulation \simeq_{Tr}^S can also be extended to open indexed processes.

Definition 12. For two open indexed triggered processes M, N , we write $M \simeq_{Tr}^S N$ if $M\{\tilde{K}/\tilde{X}\} \simeq_{Tr}^S N\{\tilde{K}/\tilde{X}\}$ for any \tilde{K} , where $\tilde{X} = fv(M) \cup fv(N)$.

3.4. The equivalence between indexed bisimulations

Now we study the relations between the three indexed bisimulations. The main result is summarized in [Proposition 8](#): $M \simeq_{Nr}^S N \Leftrightarrow Tr^n[M] \simeq_{Tr}^{S \cup \{n\}} Tr^n[N] \Leftrightarrow M \simeq_{Ct}^S N$. We achieve this result by proving several propositions including indexed factorization theorem ([Proposition 4](#)), full abstraction of the mapping to indexed triggered processes ([Proposition 5](#)), the relation between indexed triggered bisimulation and indexed normal bisimulation ([Proposition 6](#)), and the relation between indexed triggered bisimulation and indexed context bisimulation ([Proposition 7](#)).

In the following, we first give congruence of \simeq_{Ct}^S and \simeq_{Tr}^S .

Proposition 1 (Congruence of \simeq_{Ct}^S). For all $M, N, K \in IPr$, $M \simeq_{Ct}^S N$ implies:

1. $I\pi.M \simeq_{Ct}^S I\pi.N$;
2. $M|K \simeq_{Ct}^S N|K$;
3. $(va)M \simeq_{Ct}^S (va)N$;
4. $!M \simeq_{Ct}^S !N$;
5. $\{\bar{a}\langle M \rangle\}_i.K \simeq_{Ct}^S \{\bar{a}\langle N \rangle\}_i.K$.

Proof. The proof strategy is to prove that $R = \{C[M], C[N] : M \simeq_{Ct}^S N\}$ is contained in \simeq_{Ct}^S , where $C[\]$ is in one of the forms $I\pi.[\]$, $[\]|K$, $(va)[\]$, $![\]$, and $\{\bar{a}\langle [\] \rangle\}_i.K$. The proof is similar to the argument of the analogous result for context bisimulation in [16, Theorem 4.2.7]. The only difference between the proof in [16, Theorem 4.2.7] is that in this case every process and every action are assigned indices. \square

Proposition 2 (Congruence of \simeq_{Tr}^S). For all $M, N, K \in IPr$, $M \simeq_{Tr}^S N$ implies:

1. $M|K \simeq_{Tr}^S N|K$;
2. $(\nu a)M \simeq_{Tr}^S (\nu a)N$.

Proof. The proof strategy is to prove that $R = \{(C[M], C[N]) : M \simeq_{Tr}^S N\}$ is contained in \simeq_{Tr}^S , where $C[\]$ is in one of the forms $[\]|K$ and $(\nu a)[\]$. The proof is similar to the argument of the analogous result for triggered bisimulation in [16, Lemma 4.6.3]. The only difference between the proof in [16, Lemma 4.6.3] is that in this case every process and every action are assigned indices. \square

Proposition 3 states the easy part of the relation between \simeq_{Ct}^S and \simeq_{Nr}^S .

Proposition 3. For any $M, N \in IPr^c$, $M \simeq_{Ct}^S N \Rightarrow M \simeq_{Nr}^S N$.

Proof. It is trivial by the definition of \simeq_{Ct}^S and \simeq_{Nr}^S since the requirements in the definition of \simeq_{Nr}^S are a subset of those in the definition of \simeq_{Ct}^S . \square

Now we give the indexed version of the factorization theorem, which states that, by means of indexed triggers, an indexed subprocess of a given indexed process can be factorized out.

Proposition 4. For any indexed processes M and N with $m \notin \text{fn}(M, N)$, it holds that $M\{\{\tau\}_{i,j}.N/U\} \simeq_{Ct}^S (\nu m)(M\{\{\bar{m}\}_i.0/U\}|\{m\}_j.N)$ for any S .

Proof. Similar to the proof of $P\{\tau.R/X\} \sim_{Ct} (\nu m)(P\{\bar{m}.0/X\}|\{m\}.R)$ in [16] except that every prefix is assigned an index, by induction on the structure of M . \square

Corollary 1. For any indexed processes M and N with $m \notin \text{fn}(M, N)$, it holds that $M\{N/U\} \simeq_{Ct}^{S \cup \{n\}} (\nu m)(M\{\{\bar{m}\}_n.0/U\}|\{m\}_n.N)$ for any S .

Proof. It is straightforward by $\{\tau\}_{n,n}.M \simeq_{Ct}^{S \cup \{n\}} M$ and Propositions 1 and 4. \square

To prove the correctness of $Tr^n[\]$, which is stated as Proposition 5, we first give the following lemma:

Lemma 1. For any $M, N \in IPr^c$, $M \simeq_{Ct}^S N \Rightarrow M \simeq_{Tr}^S N$.

Proof. See Appendix A. \square

In the proof of Lemma 1, we need a technical result: $(\nu m)(\nu \tilde{b})(M|\{m\}_n.K)|C\{\{\bar{m}\}_n.0\} \simeq_{Ct}^S (\nu m)(\nu \tilde{c})(N|\{m\}_n.L)|C\{\{\bar{m}\}_n.0\}$ implies $(\nu \tilde{b})(M|\{m\}_n.K) \simeq_{Ct}^S (\nu \tilde{c})(N|\{m\}_n.L)$. In fact, this result can be achieved by proving that $(\nu m)((\nu \tilde{b})(M|\{m\}_n.K)|\{m'\}_n.\{\bar{m}\}_n.0) \simeq_{Ct}^S (\nu m)((\nu \tilde{c})(N|\{m\}_n.L)|\{m'\}_n.\{\bar{m}\}_n.0)$ implies $(\nu \tilde{b})(M|\{m'\}_n.K) \simeq_{Ct}^S (\nu \tilde{c})(N|\{m'\}_n.L)$, which is stated as Lemma A.2 in Appendix A.

The following proposition states the correctness of mapping $Tr^n[\]$.

Proposition 5. For each $M \in IPr^c$,

1. $Tr^n[M]$ is an indexed triggered process;
2. $Tr^n[M] \simeq_{Ct}^{S \cup \{n\}} M$;
3. $Tr^n[M] \simeq_{Tr}^{S \cup \{n\}} M$, if M is an indexed triggered process.

Proof. 1. It is straightforward.

2. Since an indexed triggered process is also an indexed process, Corollary 1 holds for indexed triggered processes. This claim can be proved by induction on the structure of M and using Corollary 1.

3. Since an indexed triggered process is always an indexed process, by Claim 1, $Tr^n[M]$ is still an indexed triggered process for indexed triggered process M . By Lemma 1 and Claim 2, the claim holds. \square

The following lemma gives the operational correspondence for $Tr^n[\]$ with respect to indexed triggered bisimulation, which is an indexed version of Lemma 7.2 in [17]. This lemma is used to study the relation between \simeq_{Nr}^S and $\simeq_{Tr}^{S \cup \{n\}}$ in Proposition 6.

Lemma 2. Let $m \notin \text{fn}(M, K)$ and $n \notin \text{ix}(M, K)$, where M and K are indexed processes.

1. (a) If $M \xrightarrow{\{(\nu \tilde{b})\bar{a}(K)\}_i.S} M'$, then $Tr^n[M] \xrightarrow{\{(\nu m)\bar{a}(\{\bar{m}\}_n.0)\}_i.S \cup \{n\}} M''$ and $M'' \simeq_{Tr}^{S \cup \{n\}} Tr^n[(\nu \tilde{b})(M'|\{m\}_n.K)]$;
- (b) if $M \xrightarrow{\{a(\{\bar{m}\}_j.0)\}_i.S} M'$, then $Tr^n[M] \xrightarrow{\{a(\{\bar{m}\}_j.0)\}_i.S \cup \{n\}} M''$ and $M'' \simeq_{Tr}^{S \cup \{n\}} Tr^n[M']$;
- (c) suppose $I\alpha$ is not an indexed higher order action and $I\alpha \neq \{\tau\}_{i,i}$ for any $i \in S$; if $M \xrightarrow{I\alpha.S} M'$, then M'' exists such that $Tr^n[M] \xrightarrow{I\alpha.S \cup \{n\}} M''$ and $M'' \simeq_{Tr}^{S \cup \{n\}} Tr^n[M']$;
- (d) if $M \xrightarrow{\varepsilon.S} M'$, then M'' exists such that $Tr^n[M] \xrightarrow{\varepsilon.S \cup \{n\}} M''$ and $M'' \simeq_{Tr}^{S \cup \{n\}} Tr^n[M']$.
2. (a) If $Tr^n[M] \xrightarrow{\{(\nu m)\bar{a}(\{\bar{m}\}_n.0)\}_i.S \cup \{n\}} M''$, then M', K and \tilde{b} exist such that $M \xrightarrow{\{(\nu \tilde{b})\bar{a}(K)\}_i.S} M'$ and $M'' \simeq_{Tr}^{S \cup \{n\}} Tr^n[(\nu \tilde{b})(M'|\{m\}_n.K)]$;

- (b) if $\text{Tr}^n[M] \xrightarrow{\{a(\{\bar{m}\}_j.0)\}_i, S \cup \{n\}} M''$, then $M \xrightarrow{\{a(\{\bar{m}\}_j.0)\}_i, S} M'$ and $M'' \simeq_{\text{Tr}}^{S \cup \{n\}} \text{Tr}^n[M']$;
- (c) suppose $\text{I}\alpha$ is not an indexed higher order action and $\text{I}\alpha \neq \{\tau\}_{i,i}$ for any $i \in S \cup \{n\}$; if $\text{Tr}^n[M] \xrightarrow{\text{I}\alpha, S \cup \{n\}} M''$, then M' exists such that $M \xrightarrow{\text{I}\alpha, S} M'$ and $M'' \simeq_{\text{Tr}}^{S \cup \{n\}} \text{Tr}^n[M']$;
- (d) if $\text{Tr}^n[M] \xrightarrow{\varepsilon, S \cup \{n\}} M''$, then M' exists such that $M \xrightarrow{\varepsilon, S} M'$ and $M'' \simeq_{\text{Tr}}^{S \cup \{n\}} \text{Tr}^n[M']$.

Proof. By induction on the length of transition, similar to the proof of Lemma 4.7.3 in [16]. The only difference between the proof in [16, Lemma 4.7.3] is that in this case every process and every action are assigned indices. \square

For an example of higher order output case of Lemma 2, we have $\{\bar{a}\langle N \rangle\}_i.L \xrightarrow{\{\bar{a}\langle N \rangle\}_i, \emptyset} L$, $\text{Tr}^n[\{\bar{a}\langle N \rangle\}_i.L] \equiv (\nu m)(\{\bar{a}\langle \{\bar{m}\}_n.0 \rangle\}_i.\text{Tr}^n[L]!\{m\}_n.\text{Tr}^n[N]) \xrightarrow{\{(\nu m)\bar{a}\langle \{\bar{m}\}_n.0 \rangle\}_i, \{n\}} \text{Tr}^n[L]!\{m\}_n.\text{Tr}^n[N]$, and $\text{Tr}^n[L]!\{m\}_n.\text{Tr}^n[N] \simeq_{\text{Tr}}^{\{n\}} \text{Tr}^n[L]!\{m\}_n.N$.

For an example of higher order input case of Lemma 2, we have $\{a(U)\}_i.(U|U) \xrightarrow{\{a(\{\bar{m}\}_j.0)\}_i, \emptyset} \{\bar{m}\}_j.0|\{\bar{m}\}_j.0$, $\text{Tr}^n[\{a(U)\}_i.(U|U)] \equiv \{a(U)\}_i.(U|U) \xrightarrow{\{a(\{\bar{m}\}_j.0)\}_i, \{n\}} \{\bar{m}\}_j.0|\{\bar{m}\}_j.0$, and $\{\bar{m}\}_j.0|\{\bar{m}\}_j.0 \simeq_{\text{Tr}}^{\{n\}} \{\bar{m}\}_j.0|\{\bar{m}\}_j.0$.

Proposition 6 below states the relation between \simeq_{Nr}^S and $\simeq_{\text{Tr}}^{S \cup \{n\}}$:

Proposition 6. For any $M, N \in \text{IPr}^c$, $M \simeq_{\text{Nr}}^S N \Rightarrow \text{Tr}^n[M] \simeq_{\text{Tr}}^{S \cup \{n\}} \text{Tr}^n[N]$, where $n \notin \text{ix}(M, N)$.

Proof. See Appendix B. \square

The following lemma gives the operational correspondence for $\text{Tr}^n[\]$ with respect to indexed context bisimulation, which is an indexed version of Lemma 5.5 in [17]. This lemma is used to study the relation between \simeq_{Tr}^S and \simeq_{Ct}^S in Lemma 4.

Lemma 3. Let $m \notin \text{fn}(M, K)$ and $n \notin \text{ix}(M, K)$.

- (a) If $M \xrightarrow{\{a(K)\}_i, S} M'$, then M'' exists such that $\text{Tr}^n[M] \xrightarrow{\{a(\{\bar{m}\}_n.0)\}_i, S \cup \{n\}} M''$ and $\text{Tr}^n[M'] \simeq_{\text{Ct}}^{S \cup \{n\}} (\nu m)(M''!\{m\}_n.K)$;
- (b) if $M \xrightarrow{\{(\nu \tilde{b})\bar{a}\langle K \rangle\}_i, S} M'$, then M'' exists such that $\text{Tr}^n[M] \xrightarrow{\{(\nu m)\bar{a}\langle \{\bar{m}\}_n.0 \rangle\}_i, S \cup \{n\}} M''$ and $\text{Tr}^n[(\nu \tilde{b})(M'|C\langle K \rangle)] \simeq_{\text{Ct}}^{S \cup \{n\}} (\nu m)(M''!\text{Tr}^n[C\langle \{\bar{m}\}_n.0 \rangle])$ for any $C(U)$ with $\text{fn}(C(U)) \cap \{\tilde{b}, m\} = \emptyset$;
- (c) suppose $\text{I}\alpha$ is not an indexed higher order action and $\text{I}\alpha \neq \{\tau\}_{i,i}$ for any $i \in S$; if $M \xrightarrow{\text{I}\alpha, S} M'$, then M'' exists such that $\text{Tr}^n[M] \xrightarrow{\text{I}\alpha, S \cup \{n\}} M''$ and $M'' \simeq_{\text{Ct}}^{S \cup \{n\}} \text{Tr}^n[M']$;
- (d) if $M \xrightarrow{\varepsilon, S} M'$, then M'' exists such that $\text{Tr}^n[M] \xrightarrow{\varepsilon, S \cup \{n\}} M''$ and $M'' \simeq_{\text{Ct}}^{S \cup \{n\}} \text{Tr}^n[M']$.
- (a) If $\text{Tr}^n[M] \xrightarrow{\{a(\{\bar{m}\}_n.0)\}_i, S \cup \{n\}} M''$, then M' exists such that $M \xrightarrow{\{a(K)\}_i, S} M'$ and $\text{Tr}^n[M'] \simeq_{\text{Ct}}^{S \cup \{n\}} (\nu m)(M''!\{m\}_n.K)$;
- (b) if $\text{Tr}^n[M] \xrightarrow{\{(\nu \tilde{b})\bar{a}\langle \{\bar{m}\}_n.0 \rangle\}_i, S \cup \{n\}} M''$, then M' exists such that $M \xrightarrow{\{(\nu \tilde{b})\bar{a}\langle K \rangle\}_i, S} M'$ and $\text{Tr}^n[(\nu \tilde{b})(M'|C\langle K \rangle)] \simeq_{\text{Ct}}^{S \cup \{n\}} (\nu m)(M''!\text{Tr}^n[C\langle \{\bar{m}\}_n.0 \rangle])$ for any $C(U)$ with $\text{fn}(C(U)) \cap \{\tilde{b}, m\} = \emptyset$;
- (c) suppose $\text{I}\alpha$ is not an indexed higher order action and $\text{I}\alpha \neq \{\tau\}_{i,i}$ for any $i \in S \cup \{n\}$; if $\text{Tr}^n[M] \xrightarrow{\text{I}\alpha, S \cup \{n\}} M''$, then M' exists such that $M \xrightarrow{\text{I}\alpha, S} M'$ and $M'' \simeq_{\text{Ct}}^{S \cup \{n\}} \text{Tr}^n[M']$;
- (d) if $\text{Tr}^n[M] \xrightarrow{\varepsilon, S \cup \{n\}} M''$, then M' exists such that $M \xrightarrow{\varepsilon, S} M'$ and $M'' \simeq_{\text{Ct}}^{S \cup \{n\}} \text{Tr}^n[M']$.

Proof. By induction on the length of transition, similar to the proofs of Corollary 4.4.8 and Lemma 4.7.3 in [16] except that in this case every process and every action are assigned indices. \square

For an example of higher order output case of Lemma 3, we have $\{\bar{a}\langle N \rangle\}_i.L \xrightarrow{\{\bar{a}\langle N \rangle\}_i, \emptyset} L$, $\text{Tr}^n[\{\bar{a}\langle N \rangle\}_i.L] \equiv (\nu m)(\{\bar{a}\langle \{\bar{m}\}_n.0 \rangle\}_i.\text{Tr}^n[L]!\{m\}_n.\text{Tr}^n[N]) \xrightarrow{\{(\nu m)\bar{a}\langle \{\bar{m}\}_n.0 \rangle\}_i, \{n\}} \text{Tr}^n[L]!\{m\}_n.\text{Tr}^n[N]$, and $(\nu m)(\text{Tr}^n[L]!\{m\}_n.\text{Tr}^n[N]|\text{Tr}^n[C\langle \{\bar{m}\}_n.0 \rangle]) \simeq_{\text{Tr}}^{\{n\}} \text{Tr}^n[L]|\text{Tr}^n[C\langle N \rangle]$ for any $C(U)$ with $\text{fn}(C(U)) \cap \{m\} = \emptyset$.

For an example of higher order input case of Lemma 3, we have $\{a(U)\}_i.(U|U) \xrightarrow{\{a(K)\}_i, \emptyset} K|K$, $\text{Tr}^n[\{a(U)\}_i.(U|U)] \equiv \{a(U)\}_i.(U|U) \xrightarrow{\{a(\{\bar{m}\}_j.0)\}_i, \{n\}} \{\bar{m}\}_j.0|\{\bar{m}\}_j.0$, and $\text{Tr}^n[K|K] \simeq_{\text{Ct}}^{\{n\}} (\nu m)(\{\bar{m}\}_j.0|\{\bar{m}\}_j.0|\{m\}_n.K)$.

The following Lemmas 4 and 5 are necessary to the proof of Proposition 7.

Lemma 4. For any $M, N \in \text{IPr}^c$, $\text{Tr}^n[M] \simeq_{\text{Tr}}^{S \cup \{n\}} \text{Tr}^n[N] \Rightarrow M \simeq_{\text{Ct}}^{S \cup \{n\}} N$, where $n \notin \text{ix}(M, N)$.

Proof. See Appendix C. \square

Lemma 5. For any $M, N \in \text{IPr}^c$, $M \simeq_{\text{Ct}}^{S \cup \{n\}} N \Rightarrow M \simeq_{\text{Ct}}^S N$, where $n \notin \text{ix}(M, N)$.

Proof. It is clear since $n \notin \text{ix}(M, N)$. \square

Now we get the relation between $\simeq_{\text{Tr}}^{S \cup \{n\}}$ and \simeq_{Ct}^S as follows:

Proposition 7. For any $M, N \in \text{IPr}^c$, $\text{Tr}^n[M] \simeq_{\text{Tr}}^{S \cup \{n\}} \text{Tr}^n[N] \Rightarrow M \simeq_{\text{Ct}}^S N$, where $n \notin \text{ix}(M, N)$.

Proof. By Lemmas 4 and 5. \square

The following proposition is the main result of this section, which states the equivalence between indexed context bisimulation, indexed normal bisimulation and indexed triggered bisimulation.

Proposition 8. For any $M, N \in IPr^c$, $M \simeq_{Nr}^S N \Leftrightarrow Tr^n[M] \simeq_{Tr}^{S \cup \{n\}} Tr^n[N] \Leftrightarrow M \simeq_{Ct}^S N$, where $n \notin ix(M, N)$.

Proof. By Propositions 3, 6 and 7. \square

For indexed triggered processes, the above proposition can be simplified as Corollary 2.

Lemma 6. For any $M, N \in ITPr^c$, $M \simeq_{Tr}^{S \cup \{n\}} N \Rightarrow M \simeq_{Tr}^S N$, where $n \notin ix(M, N)$.

Proof. It is clear since $n \notin ix(M, N)$. \square

Corollary 2. For any $M, N \in ITPr^c$, $M \simeq_{Nr}^S N \Leftrightarrow M \simeq_{Tr}^S N \Leftrightarrow M \simeq_{Ct}^S N$.

Proof. By Proposition 8, $M \simeq_{Nr}^S N \Leftrightarrow Tr^n[M] \simeq_{Tr}^{S \cup \{n\}} Tr^n[N] \Leftrightarrow M \simeq_{Ct}^S N$, where $n \notin ix(M, N)$. Since $M, N \in ITPr^c$, $M \simeq_{Tr}^{S \cup \{n\}} Tr^n[M] \simeq_{Tr}^{S \cup \{n\}} Tr^n[N] \simeq_{Tr}^{S \cup \{n\}} N$. By Lemma 4, we have $M \simeq_{Tr}^S N$. \square

Sangiorgi [16] proved that barbed congruence coincides with context bisimulation. We generalize this result to our indexed process calculus. In the following, we firstly present an indexed variant of barbed congruence. Then we show that this indexed barbed congruence coincides with an indexed version of reduction congruence called indexed reduction congruence. Moreover, we give the equivalence between indexed reduction congruence, indexed context bisimulation and indexed normal bisimulation. This result shows that all our indexed bisimulations are same and capture the essential of equivalence of indexed processes.

Definition 13. For each name a , the observability predicate \downarrow_a is defined by

$$M \downarrow_a \text{ if } \exists M', M \xrightarrow{I\alpha} M', \text{ where } I\alpha = \{a\langle K \rangle\}_i \text{ or } \{(\nu \tilde{b})\bar{a}\langle K \rangle\}_i.$$

Definition 14. $M \Downarrow_a^S$ means $\exists M', M \xRightarrow{\varepsilon, S} M' \downarrow_a$.

Definition 15. Let M, N be two indexed processes, and $S \subseteq I$ be an index set, we write $M \simeq_{Ba}^S N$, if there is a symmetric relation R and $M R N$ implies:

- (1) $M|K R N|K$ for any indexed process K ;
- (2) whenever $M \xRightarrow{\varepsilon, S} M'$, there exists N' such that $N \xRightarrow{\varepsilon, S} N'$ and $M' R N'$;
- (3) whenever $M \xRightarrow{\{\tau\}_{i,j}, S} M'$, where $(i, j) \notin \{(k, k) | k \in S\}$, there exists N' such that $N \xRightarrow{\{\tau\}_{i,j}, S} N'$ and $M' R N'$;
- (4) if $M \Downarrow_a^S$ then $N \Downarrow_a^S$.

We say that M and N are indexed barbed congruence with respect to S if $M \simeq_{Ba}^S N$. Since \simeq_{Ct}^S is equivalent to \simeq_{Nr}^S .

The following proposition states that \simeq_{Ct}^S , \simeq_{Nr}^S and \simeq_{Ba}^S are same.

Claim (3) is necessary for \simeq_{Ba}^S . If Claim (3) is not included in the above definition of \simeq_{Ba}^S , \simeq_{Ba}^S will not be equivalent to \simeq_{Ct}^S . For example, it is clear $!\{\bar{a}\langle 0 \rangle.0\}_i \not\approx_{Ct}^{i,j} !\{\bar{a}\langle 0 \rangle.0\}_j$, where $i \neq j$. But it is easy to see that $R = \{(!\{\bar{a}\langle 0 \rangle.0\}_i|K, !\{\bar{a}\langle 0 \rangle.0\}_j|K) : K \text{ is an arbitrary indexed process}\}$ satisfies conditions (1), (2) and (4). So Claim (3) should be included in the definition of \simeq_{Ba}^S .

On the other hand, Claim (4) is not necessary. In the following, we will present an indexed reduction-based bisimulation without barbed testing and prove the equivalence between such bisimulation and \simeq_{Ba}^S .

Definition 16. Let M, N be two indexed processes, and $S \subseteq I$ be an index set, we write $M \simeq_{Rd}^S N$, if there is a symmetric relation R and $M R N$ implies:

- (1) $M|K R N|K$ for any indexed process K ;
- (2) whenever $M \xRightarrow{\varepsilon, S} M'$, there exists N' such that $N \xRightarrow{\varepsilon, S} N'$ and $M' R N'$;
- (3) whenever $M \xRightarrow{\{\tau\}_{i,j}, S} M'$, where $(i, j) \notin \{(k, k) | k \in S\}$, there exists N' such that $N \xRightarrow{\{\tau\}_{i,j}, S} N'$ and $M' R N'$.

We say that M and N are indexed reduction congruence with respect to S if $M \simeq_{Rd}^S N$. Since \simeq_{Ct}^S is equivalent to \simeq_{Nr}^S , the following proposition states that \simeq_{Ct}^S , \simeq_{Nr}^S and \simeq_{Rd}^S are same.

The concept of indexed processes are not only used for the proof of equivalence of context bisimulation and normal bisimulation, but also can be regarded as a kind of “distributed processes”. For example, $\{\bar{a}\}_i.0|\{\bar{b}\}_j.0|\{a\}_m.\{b\}_n.0$ denotes a “distributed process” which is able to perform an output action \bar{a} at location i , and perform an output action \bar{b} at location j , and perform an input action a at location m , and perform an input action b at location n . Although the conceptions of indexed barbed congruence and indexed reduction congruence are not necessary for the proof of equivalence of context bisimulation and normal bisimulation, we presented and studied these indexed congruences for they can be regarded as a kind of equivalence for “distributed processes”.

We can see that barbed equivalence and indexed reduction congruence are both reduction-based bisimulations on one hand. On the other hand, for barbed equivalence, an observer can test channels through which process can interact with environment, whereas for indexed reduction congruence, an observer need not test channels but has capability to distinguish components where communication happens. Therefore our indexed reduction congruence is not a direct generalization of barbed equivalence.

Proposition 9. For any $M, N \in \text{IPr}^c$, $M \simeq_{\text{Ct}}^S N \Leftrightarrow M \simeq_{\text{Ba}}^S N \Leftrightarrow M \simeq_{\text{Rd}}^S N \Leftrightarrow M \simeq_{\text{Nr}}^S N$.

Proof. See Appendix D. \square

In [16], Sangiorgi showed that reduction congruence is not equivalent to barbed congruence. But for our indexed process calculus, the above proposition shows that the indexed reduction congruence is equivalent to indexed barbed congruence. This gives the essential difference between indexed processes and original processes.

Barbed congruence is regarded as a uniform definition of bisimulation for a variety of concurrency calculi, such as π -calculus [5,16], higher order π -calculus [16], Join calculus [6], and safe ambient calculus [12]. But the definition of barb will vary with different calculi. For example, definitions of barb are different for π -calculus and safe ambient calculus. Indexed reduction congruence can be also used as a uniform equivalence for different process calculi. In [1], the concept of indexed reduction congruence was used to give a uniform equivalence for π -calculus, higher order π -calculus, and safe ambient calculus.

4. The equivalence between bisimulations in higher order π -calculus

4.1. Strong context bisimulation coincides with strong normal bisimulation

The equivalence between strong context bisimulation and strong normal bisimulation can be derived by the mapping to indexed triggered process and the equivalence between indexed bisimulations.

For example, let us see the following two processes:

$$P = (\nu a)(\bar{a}(\bar{b}.0).0|a(X).X);$$

$$Q = (\nu a)(\bar{a}(0).0|a(X).\bar{b}.0).$$

They are clearly strongly context bisimilar. However, their triggered mappings are not strongly triggered bisimilar. Indeed, the mapping of Q is $(\nu a)((\nu m)(\bar{a}(\bar{m}.0).0|!m.0)|a(X).\bar{b}.0)$, after the communication between \bar{a} and a , the residual process can perform action \bar{b} without using silent tau actions, whereas the mapping of P is $(\nu a)((\nu m)(\bar{a}(\bar{m}.0).0|!m.\bar{b}.0)|a(X).X)$, and to match this behaviour, one has to go through a trigger and this therefore requires abstracting of some tau actions. Hence the proof strategy in [16,17] cannot be generalized to the case of strong bisimulation.

In our approach, we firstly consider the indexed version of P and Q :

$$\{P\}_0 = (\nu a)((\bar{a}(\{\bar{b}\}_0.0))_0.0|a(X)_0.X);$$

$$\{Q\}_0 = (\nu a)((\bar{a}(0))_0.0|a(X)_0.\{\bar{b}\}_0.0).$$

It is clear that $\{P\}_0 \simeq_{\text{Ct}}^{\emptyset} \{Q\}_0$. Now the indexed triggered mapping of $\{P\}_0$ is $\text{Tr}^n[\{P\}_0] = (\nu a)((\nu m)((\bar{a}(\{\bar{m}\}_n.0))_0.0|!m_n.\{\bar{b}\}_0.0)|a(X)_0.X)$, and the indexed triggered mapping of $\{Q\}_0$ is $\text{Tr}^n[\{Q\}_0] = (\nu a)((\nu m)((\bar{a}(\{\bar{m}\}_n.0))_0.0|!m_n.0)|a(X)_0.\{\bar{b}\}_0.0)$. Unlike the un-indexed case, $\text{Tr}^n[\{P\}_0]$ and $\text{Tr}^n[\{Q\}_0]$ are indexed triggered bisimilar with respect to $S = \{n\}$. For example, let us consider the transition: $\text{Tr}^n[\{P\}_0] \xrightarrow{\{\tau\}_{0,0}} (\nu a)((\nu m)(0|!m_n.\{\bar{b}\}_0.0|(\bar{a}(\{\bar{m}\}_n.0))_0.0) \xrightarrow{\{\tau\}_{n,n}} (\nu a)((\nu m)(0|(\{\bar{b}\}_0.0|!m_n.\{\bar{b}\}_0.0|0)) \xrightarrow{\{\bar{b}\}_0} (\nu a)((\nu m)(0|0|!m_n.\{\bar{b}\}_0.0|0))$. Since we abstract indexed tau actions of the form $\{\tau\}_{n,n}$ in the definition of $\simeq_{\text{Tr}}^{(n)}$, we have a matching transition $\text{Tr}^n[\{Q\}_0] \xrightarrow{\{\tau\}_{0,0}} (\nu a)((\nu m)(0|!m_n.0|(\bar{a}(\{\bar{m}\}_n.0))_0.0) \xrightarrow{\{\bar{b}\}_0} (\nu a)((\nu m)(0|!m_n.0|0))$. Hence $\text{Tr}^n[\{P\}_0]$ and $\text{Tr}^n[\{Q\}_0]$ are bisimilar. Formally, we have $\text{Tr}^n[\{P\}_0] \simeq_{\text{Tr}}^{(n)} \text{Tr}^n[\{Q\}_0]$. Similarly, we can further build the relation between $\simeq_{\text{Tr}}^{(n)}$ and $\simeq_{\text{Nr}}^{\emptyset}$: $\text{Tr}^n[\{P\}_0] \simeq_{\text{Tr}}^{(n)} \text{Tr}^n[\{Q\}_0] \Leftrightarrow \{P\}_0 \simeq_{\text{Nr}}^{\emptyset} \{Q\}_0$.

In this section, we will show that $P \sim_{\text{Nr}} Q \Rightarrow \{P\}_0 \simeq_{\text{Nr}}^{\emptyset} \{Q\}_0$ and $\{P\}_0 \simeq_{\text{Ct}}^{\emptyset} \{Q\}_0 \Rightarrow P \sim_{\text{Ct}} Q \Rightarrow P \sim_{\text{Nr}} Q$. Since $\{P\}_0 \simeq_{\text{Ct}}^{\emptyset} \{Q\}_0 \Leftrightarrow \{P\}_0 \simeq_{\text{Nr}}^{\emptyset} \{Q\}_0$ by Proposition 8, the equivalence between $P \sim_{\text{Nr}} Q$ and $P \sim_{\text{Ct}} Q$ is obvious.

Now we prove that strong context bisimulation and strong normal bisimulation coincide, which was presented in [17] as an open problem.

Firstly, we introduce the concept of strong indexed context equivalence, strong indexed normal equivalence and strong indexed triggered equivalence.

Definition 17 (Strong Indexed Context Equivalence). Let $P, Q \in \text{Pr}^c$, we write $P \sim_{\text{Ct}}^i Q$, if $\{P\}_k \simeq_{\text{Ct}}^{\emptyset} \{Q\}_k$ for some index k . As we defined before, where $\{P\}_k$ denotes indexed process with the same given index k on every prefix in P .

Definition 18 (Strong Indexed Normal Equivalence). Let $P, Q \in \text{Pr}^c$, we write $P \sim_{\text{Nr}}^i Q$, if $\{P\}_k \simeq_{\text{Nr}}^{\emptyset} \{Q\}_k$ for some index k .

Definition 19 (Strong Indexed Triggered Equivalence). Let $P, Q \in \text{Pr}^c$, we write $P \sim_{\text{Tr}}^{i,(n)} Q$, if $\text{Tr}^n[\{P\}_k] \simeq_{\text{Tr}}^{(n)} \text{Tr}^n[\{Q\}_k]$ for some index k with $k \neq n$.

The following lemma states that strong normal bisimulation implies strong indexed normal equivalence.

Lemma 7. For any $P, Q \in \text{Pr}^c$, $P \sim_{\text{Nr}} Q \Rightarrow P \sim_{\text{Nr}}^i Q$.

Proof. See Appendix E. \square

Now, the equivalence between \sim_{Nr} and \sim_{Ct} can be given.

Proposition 10. For any $P, Q \in Pr^c$ and any index n , $P \sim_{Nr} Q \Leftrightarrow P \sim_{Nr}^i Q \Leftrightarrow P \sim_{Tr}^{i, \{n\}} Q \Leftrightarrow P \sim_{Ct}^i Q \Leftrightarrow P \sim_{Ct} Q$.

Proof. Firstly, by the definition, it is easy to prove $P \sim_{Ct} Q \Rightarrow P \sim_{Nr} Q$ since the requirements in the definition of \sim_{Nr} are a subset of those in the definition of \sim_{Ct} .

Furthermore, since $P \sim_{Ct}^i Q$ means $\{P\}_k \simeq_{Ct}^{\emptyset} \{Q\}_k$ for some index k , to prove $P \sim_{Ct}^i Q \Rightarrow P \sim_{Ct} Q$, we only need to prove $\{P\}_k \simeq_{Ct}^{\emptyset} \{Q\}_k \Rightarrow P \sim_{Ct} Q$. Suppose $\{P\}_k \simeq_{Ct}^{\emptyset} \{Q\}_k$ for some index k . The definition of \simeq_{Ct}^{\emptyset} is the same as the definition of \sim_{Ct} except that all processes and actions have indices. Therefore we will get $P \sim_{Ct} Q$ if we neglect all indices in the definition of \simeq_{Ct}^{\emptyset} .

By Lemma 7, $P \sim_{Nr} Q \Rightarrow P \sim_{Nr}^i Q$. Hence $P \sim_{Ct}^i Q \Rightarrow P \sim_{Ct} Q \Rightarrow P \sim_{Nr} Q \Rightarrow P \sim_{Nr}^i Q$.

By Proposition 8, we have $P \sim_{Nr}^i Q \Leftrightarrow P \sim_{Tr}^{i, \{n\}} Q \Leftrightarrow P \sim_{Ct}^i Q$ for any index n . Therefore the proposition holds. \square

Moreover, we can define strong indexed reduction equivalence \sim_{Rd}^i as follows: let $P, Q \in Pr^c$, we write $P \sim_{Rd}^i Q$, if $\{P\}_k \simeq_{Rd}^{\emptyset} \{Q\}_k$ for some index k . By Propositions 9 and 10, we know that \sim_{Rd}^i coincides with \sim_{Nr} and \sim_{Ct} .

4.2. Weak context bisimulation coincides with weak normal bisimulation

Based on the equivalence between indexed bisimulations, we can give an alternative proof for the equivalence between weak context bisimulation and weak normal bisimulation.

Definition 20. (Weak Indexed Context Equivalence) Let $P, Q \in Pr^c$, we write $P \approx_{Ct}^I Q$, if $\{P\}_k \simeq_{Ct}^I \{Q\}_k$ for some index k , where I is the full index set.

Definition 21. (Weak Indexed Normal Equivalence) Let $P, Q \in Pr^c$, we write $P \approx_{Nr}^I Q$, if $\{P\}_k \simeq_{Nr}^I \{Q\}_k$ for some index k , where I is the full index set.

Definition 22. (Weak Indexed Triggered Equivalence) Let $P, Q \in Pr^c$, we write $P \approx_{Tr}^I Q$, if $Tr^n[\{P\}_k] \simeq_{Tr}^I Tr^n[\{Q\}_k]$ for some indices k and n , where $k \neq n$ and I is the full index set.

Lemma 8. For any $P, Q \in Pr^c$, $P \approx_{Nr} Q \Rightarrow P \approx_{Nr}^i Q$.

Proof. See Appendix F. \square

Proposition 11. For any $P, Q \in Pr^c$, $P \approx_{Nr} Q \Leftrightarrow P \approx_{Nr}^i Q \Leftrightarrow P \approx_{Tr}^i Q \Leftrightarrow P \approx_{Ct}^i Q \Leftrightarrow P \approx_{Ct} Q$.

Proof. Firstly, by the definition, it is easy to prove $P \approx_{Ct} Q \Rightarrow P \approx_{Nr} Q$ since the requirements in the definition of \approx_{Nr} are a subset of those in the definition of \approx_{Ct} .

Furthermore, $P \approx_{Ct}^i Q$ means $\{P\}_k \simeq_{Ct}^I \{Q\}_k$ for some index k . The definition of \simeq_{Ct}^I is the same as the definition of \approx_{Ct} except that all processes and actions have indices. Therefore we will get $P \approx_{Ct} Q$ if we neglect all indices in the definition of \simeq_{Ct}^I .

By Proposition 8, it is easy to get $P \approx_{Nr}^i Q \Leftrightarrow P \approx_{Tr}^i Q \Leftrightarrow P \approx_{Ct}^i Q$. By Lemma 8, $P \approx_{Nr} Q \Rightarrow P \approx_{Nr}^i Q$, therefore the proposition holds. \square

Similarly, we can define weak indexed reduction equivalence \approx_{Rd}^i as follows: let $P, Q \in Pr^c$, we write $P \approx_{Rd}^i Q$ if $\{P\}_k \simeq_{Rd}^I \{Q\}_k$ for some index k . By Propositions 9 and 11, \approx_{Rd}^i coincides with \approx_{Nr} and \approx_{Ct} .

One may doubt whether the notation $\{\tau\}_{i,j}$ (where i and j are different) is necessary for the proofs of Propositions 10 and 11, where all prefixes are given the same index k (thus “original tau actions” have subscript “ k, k ”) and all the triggers are given another index n (thus “trigger activations” have subscript “ n, n ”).

In fact, the notation $\{\tau\}_{i,j}$ (where i and j are different) is needed in the proofs of Propositions 10 and 11. For the definition of indexed context bisimulation, in the cases of higher order input and higher order output action, indexed process will input or output indexed processes. In such cases, inputted or outputted indexed processes may have some different indices, so we should introduce $\{\tau\}_{i,j}$. For example, $\{a(U).U[\bar{b}(L).0]\}_i \equiv \{a(U)\}_i.U[\bar{b}(L)]_i.0$ may receive $\{b(X)\}_k.X$ through channel a , then becomes $\{b(X)\}_k.X[\bar{b}(L)]_i.0$, and this indexed process can perform $\{\tau\}_{i,k}$.

Actually, we can replace the full set I by $\{n, k\}$ in Definition 22, and replace the full set I by $\{k\}$ in Definitions 20 and 21. Similar to Proposition 11, we can also get $P \approx_{Nr} Q \Leftrightarrow \{P\}_k \simeq_{Nr}^{\{k\}} \{Q\}_k \Leftrightarrow Tr^n[\{P\}_k] \simeq_{Tr}^{\{n,k\}} Tr^n[\{Q\}_k] \Leftrightarrow \{P\}_k \simeq_{Ct}^{\{k\}} \{Q\}_k \Leftrightarrow P \approx_{Ct} Q$. The proof strategy of this proposition is similar to that of Proposition 11.

In [16,17], the proposition: $P \approx_{Nr} Q \Leftrightarrow Tr[P] \approx_{Tr} Tr[Q] \Leftrightarrow P \approx_{Ct} Q$ was proved, where $Tr[\]$ is the triggered mapping and \approx_{Tr} is the weak triggered bisimulation. In fact, this proposition can be got from Proposition 11. Firstly by Proposition 11, we have $P \approx_{Nr} Q \Leftrightarrow Tr^n[\{P\}_k] \simeq_{Tr}^I Tr^n[\{Q\}_k] \Leftrightarrow P \approx_{Ct} Q$. Secondly, we can prove that $Tr[P] \approx_{Tr} Tr[Q] \Leftrightarrow Tr^n[\{P\}_k] \simeq_{Tr}^I Tr^n[\{Q\}_k]$. Hence $P \approx_{Nr} Q \Leftrightarrow Tr[P] \approx_{Tr} Tr[Q] \Leftrightarrow P \approx_{Ct} Q$ is a corollary of Proposition 11. But for the strong case, the claim: $P \approx_{Nr} Q \Leftrightarrow Tr[P] \approx_{Tr} Tr[Q] \Leftrightarrow P \approx_{Ct} Q$ does not hold. For example, let $P = (\nu a)(\bar{a}(b.0).0)a(X).X$ and $Q = (\nu a)(\bar{a}(0).0)a(X).b.0$, then $P \approx_{Nr} Q$, $P \approx_{Ct} Q$ and $Tr[P] \not\approx_{Tr} Tr[Q]$. Hence $P \approx_{Nr} Q \not\Rightarrow Tr[P] \approx_{Tr} Tr[Q] \not\Rightarrow P \approx_{Ct} Q$. This also shows that we cannot prove the equivalence between strong context bisimulation and strong normal bisimulation by the original technique of triggered mapping.

4.3. A variant of normal bisimulation

Let us see the language of higher order π -calculus without replication operators defined by the following grammar:

$$\begin{aligned} P &::= 0 \mid U \mid \pi.P \mid P_1 \mid P_2 \mid (\nu a)P \\ \pi &::= \tau \mid l \mid \bar{l} \mid a(U) \mid \bar{a}(P). \end{aligned}$$

We write the set of processes of this higher order π -calculus as Pr_p .

In [15], Parrow has shown that in higher order π -calculus, replication can be defined by inaction, prefix, sum, parallel and restriction up to weak bisimulations. For example, $!P$ can be simulated by $R_P = (\nu a)(D \mid \bar{a}(P \mid D))$, where $D = a(X).(X \mid \bar{a}(X))$. Hence the expressive power of Pr_p is equivalent to whole higher order π -calculus. In fact, some recent higher order process calculi do not contain replication [7,8,19]. But what kind of bisimulation can be defined on Pr_p ? In the definition of normal bisimulation, replication appears. Although we can translate replication into other operators, the form is somewhat complicated. Therefore normal bisimulation seems unsuitable for Pr_p .

In the following, we give a variant of normal bisimulation, where parallel composition of any finitary copies is used as a limit form of replication. We prove that it coincides with normal bisimulation.

Definition 23. A symmetric relation $R \subseteq Pr^c \times Pr^c$ is a weak limit normal bisimulation if $P R Q$ implies:

- (1) whenever $P \xrightarrow{\varepsilon} P'$, there exists Q' such that $Q \xrightarrow{\varepsilon} Q'$ and $P' R Q'$;
- (2) whenever $P \xrightarrow{\alpha} P'$, there exists Q' such that $Q \xrightarrow{\alpha} Q'$, where α is in the form of l or \bar{l} , and $P' R Q'$;
- (3) whenever $P \xrightarrow{a(\bar{m}.0)} P'$, there exists Q' such that $Q \xrightarrow{a(\bar{m}.0)} Q'$ and $P' R Q'$, where m is a fresh name;
- (4) whenever $P \xrightarrow{(\nu \tilde{b})\bar{a}(E)} P'$, there exist Q', F, \tilde{c} such that $Q \xrightarrow{(\nu \tilde{c})\bar{a}(F)} Q'$ and $(\nu \tilde{b})(P' \mid \Pi_k m.E) R (\nu \tilde{c})(Q' \mid \Pi_k m.F)$ for all $k \in N = \{0, 1, 2, \dots\}$, where m is a fresh name. Here we write $\Pi_k P$ to denote the parallel composition of k copies of P , for example, $\Pi_3 P$ represents $P \mid P \mid P$.

We write $P \approx_{nor} Q$ if P and Q are weakly limit normal bisimilar.

In Claim (4) of the above definition, some additional quantification was introduced on contexts. But this definition does not contain replication operators. So this definition can be used as an equivalence for process calculus without replication operators, whereas the original normal bisimulation is unsuitable for Pr_p .

Lemma 9. $P \approx_{ct} Q \Rightarrow P \approx_{nor} Q$.

Proof. It is trivial by the definition of \approx_{ct} and \approx_{nor} since the requirements in the definition of \approx_{nor} are a subset of those in the definition of \approx_{ct} . \square

Lemma 10. $P \approx_{nor} Q \Rightarrow P \approx_{Nr} Q$.

Proof. See Appendix G. \square

Proposition 12. $P \approx_{ct} Q \Leftrightarrow P \approx_{nor} Q \Leftrightarrow P \approx_{Nr} Q$.

Proof. By the equivalence between \approx_{ct} and \approx_{Nr} , $P \approx_{Nr} Q \Leftrightarrow P \approx_{ct} Q$. Furthermore, by Lemmas 9 and 10, we have that the proposition holds. \square

Some results of context bisimulation for Pr_p can also be implied by this proposition. For example, the equivalence between early context bisimulation and late context bisimulation for Pr_p can be proved from this proposition.

The above results also hold in the strong case. The strong version of limit normal bisimulation is given as follows.

Definition 24. A symmetric relation $R \subseteq Pr^c \times Pr^c$ is a strong limit normal bisimulation if $P R Q$ implies:

- (1) whenever $P \xrightarrow{\alpha} P'$, there exists Q' such that $Q \xrightarrow{\alpha} Q'$, where α is not a higher order action and $P' R Q'$;
- (2) whenever $P \xrightarrow{a(\bar{m}.0)} P'$, there exists Q' such that $Q \xrightarrow{a(\bar{m}.0)} Q'$ and $P' R Q'$, where m is a fresh name;
- (3) whenever $P \xrightarrow{(\nu \tilde{b})\bar{a}(E)} P'$, there exist Q', F, \tilde{c} such that $Q \xrightarrow{(\nu \tilde{c})\bar{a}(F)} Q'$ and $(\nu \tilde{b})(P' \mid \Pi_k m.E) R (\nu \tilde{c})(Q' \mid \Pi_k m.F)$ for all $k \in N = \{0, 1, 2, \dots\}$, where m is a fresh name.

We write $P \sim_{nor} Q$ if P and Q are strongly limit normal bisimilar.

There is no difficulty to extend Proposition 12 to the following proposition by using the similar proof strategy except that tau action should also be matched.

Proposition 13. $P \sim_{ct} Q \Leftrightarrow P \sim_{nor} Q \Leftrightarrow P \sim_{Nr} Q$.

In [3], we gave a spatial logical characterization of context bisimulation by applying the above proposition.

5. Conclusions

This paper is an extended version of [2]. In this paper, we present a new proof technique for establishing the equivalence between strong normal and strong context bisimilarity relations in higher order π -calculus. To achieve this aim, we firstly present the concept of indexed processes and indexed triggered mapping, and then prove the equivalence between indexed normal bisimulation, indexed triggered bisimulation and indexed context bisimulation. As an application, we get the equivalence between normal bisimulation and context bisimulation for both strong and weak versions. In fact, our proof strategy is an enhancement of Sangiorgi's technique of trigger translation which facilitates a distinction between reductions in higher order π -calculus due to the presence of triggers, as opposed to ordinary reductions in higher order π -calculus. We introduce a level of indexing to tag the terms so as to allow this distinction. Many bisimulations are then defined upon the indexed terms and these are used to prove the desired correspondences.

In [16], context bisimulation and normal bisimulation were compared with barbed equivalence. In [9], authors proved a correspondence between weak normal bisimulation and a variant of barbed equivalence, called contextual barbed equivalence. In [11] an alternative proof of the correspondence between context bisimulation and barbed equivalence was given. In [10], authors proved that a restriction operator free subcalculus of higher order π -calculus is Turing-complete, and gave a complete inference system of this subcalculus. In [18], a new bisimulation was presented for higher order π -calculus and λ -calculus, and this bisimulation was proved to be equivalent to context bisimulation for higher order π -calculus. It would be interesting to understand whether our conceptions of indexed processes and indexed bisimulations can be helpful to study the relation between bisimulations in the framework of other higher order concurrency languages.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 60873025, the Natural Science Foundation of Jiangsu Province of China under Grant No. BK2008389, and the Foundation of Provincial Key Laboratory for Computer Information Processing Technology of Soochow University under Grant No. KJS0920.

Appendix A. Proof of Lemma 1

Lemma A.1. *Let M be an indexed triggered process and $M \xrightarrow{\{(vm)\bar{a}(\{\bar{m}\}_n, 0)\}_{i,S}} M'$, then for some M'', K, \tilde{b} with $m \notin \text{fn}(M'', K) \cup \{\tilde{b}\}$ we have $M' \simeq_{\text{Ct}}^S (\nu \tilde{b})(M'' \mid \{m\}_n.K)$.*

Proof. By induction on the derivation depth of the transition $M \xrightarrow{\{(vm)\bar{a}(\{\bar{m}\}_n, 0)\}_{i,S}} M'$. \square

The following Lemma A.2 is used in the proof of Lemma 1. The role of this lemma is to eliminate some factors in a bisimulation equation. An example of this lemma is in the following: let $M = (\nu m)(\nu a)(\{\bar{a}(\{\bar{b}\}_0, 0)\}_0.0 \mid \{m\}_n.\{a(X)\}_0.X \mid \{m'\}_n.\{\bar{m}\}_n.0)$, $N = (\nu m)(\nu a)(\{\bar{a}(0)\}_0.0 \mid \{m\}_n.\{a(X)\}_0.\{\bar{b}\}_0.0 \mid \{m'\}_n.\{\bar{m}\}_n.0)$. It is clear $M \simeq_{\text{Ct}}^{\{n,n\}} N$. By Lemma A.2, we get $(\nu a)(\{\bar{a}(\{\bar{b}\}_0, 0)\}_0.0 \mid \{m'\}_n.\{a(X)\}_0.X) \simeq_{\text{Ct}}^{\{n,n\}} (\nu a)(\{\bar{a}(0)\}_0.0 \mid \{m'\}_n.\{a(X)\}_0.\{\bar{b}\}_0.0)$. Furthermore, since m' is a fresh name, now changing m for m' does not affect the equivalence among processes, i.e., we have $(\nu a)(\{\bar{a}(\{\bar{b}\}_0, 0)\}_0.0 \mid \{m\}_n.\{a(X)\}_0.X) \simeq_{\text{Ct}}^{\{n,n\}} (\nu a)(\{\bar{a}(0)\}_0.0 \mid \{m\}_n.\{a(X)\}_0.\{\bar{b}\}_0.0)$. So we can eliminate factor $\{m'\}_n.\{\bar{m}\}_n.0$ in the bisimulation equation $M \simeq_{\text{Ct}}^{\{n,n\}} N$. This is used in the proof of Lemma 1 to eliminate factor $C(\{\bar{m}\}_n.0)$, which is introduced by the higher order output clause in the definition of \simeq_{Ct}^S .

We prove Lemma A.2 by the case analysis on the action in transition.

Lemma A.2. *If $m \notin \text{fn}(K, L, M, N) \cup \{\tilde{b}\} \cup \{\tilde{c}\}$, $m' \notin \text{fn}(K, L, M, N) \cup \{\tilde{b}\} \cup \{\tilde{c}\} \cup \{m\}$, then $(\nu m)((\nu \tilde{b})(M \mid \{m\}_n.K) \mid \{m'\}_n.\{\bar{m}\}_n.0) \simeq_{\text{Ct}}^S (\nu m)((\nu \tilde{c})(N \mid \{m\}_n.L) \mid \{m'\}_n.\{\bar{m}\}_n.0)$ implies $(\nu \tilde{b})(M \mid \{m'\}_n.K) \simeq_{\text{Ct}}^S (\nu \tilde{c})(N \mid \{m'\}_n.L)$.*

Proof. Let $R = \{((\nu \tilde{b})(M \mid \{m'\}_n.K), (\nu \tilde{c})(N \mid \{m'\}_n.L)) : (\nu m)((\nu \tilde{b})(M \mid \{m\}_n.K) \mid \{m'\}_n.\{\bar{m}\}_n.0) \simeq_{\text{Ct}}^S (\nu m)((\nu \tilde{c})(N \mid \{m\}_n.L) \mid \{m'\}_n.\{\bar{m}\}_n.0)\}$. It suffices to prove that R is a \simeq_{Ct}^S bisimulation.

Suppose $(\nu \tilde{b})(M \mid \{m'\}_n.K) \xrightarrow{I\alpha, S} M'$. There are two cases: $I\alpha = \{m'\}_n$ or $I\alpha \neq \{m'\}_n$.

If $I\alpha = \{m'\}_n$, since $m' \notin \text{fn}(K, L, M, N) \cup \{\tilde{b}\} \cup \{\tilde{c}\} \cup \{m\}$, we have $(\nu \tilde{b})(M \mid \{m'\}_n.K) \xrightarrow{\{m'\}_n, S} (\nu \tilde{b})(M \mid K \mid \{m'\}_n.K)$ and $(\nu \tilde{c})(N \mid \{m'\}_n.L) \xrightarrow{\{m'\}_n, S} (\nu \tilde{c})(N \mid L \mid \{m'\}_n.L)$. Furthermore, since $(\nu m)((\nu \tilde{b})(M \mid \{m\}_n.K) \mid \{m'\}_n.\{\bar{m}\}_n.0) \xrightarrow{\{m'\}_n, S} (\nu m)((\nu \tilde{b})(M \mid \{m\}_n.K \mid \{m\}_n.K) \mid \{\bar{m}\}_n.0 \mid \{m'\}_n.\{\bar{m}\}_n.0) \xrightarrow{\{\tau\}_n, S} (\nu m)((\nu \tilde{b})(M \mid K \mid \{m\}_n.K) \mid \{m'\}_n.\{\bar{m}\}_n.0)$ and $(\nu m)((\nu \tilde{b})(M \mid \{m\}_n.K) \mid \{m'\}_n.\{\bar{m}\}_n.0) \simeq_{\text{Ct}}^S (\nu m)((\nu \tilde{c})(N \mid \{m\}_n.L) \mid \{m'\}_n.\{\bar{m}\}_n.0)$, we have $(\nu m)((\nu \tilde{c})(N \mid \{m\}_n.L) \mid \{m'\}_n.\{\bar{m}\}_n.0) \xrightarrow{\{m'\}_n, S} (\nu m)((\nu \tilde{c})(N \mid \{m\}_n.L \mid \{m\}_n.L) \mid \{\bar{m}\}_n.0 \mid \{m'\}_n.\{\bar{m}\}_n.0) \xrightarrow{\{\tau\}_n, S} (\nu m)((\nu \tilde{c})(N \mid L \mid \{m\}_n.L) \mid \{m'\}_n.\{\bar{m}\}_n.0)$ and $(\nu m)((\nu \tilde{b})(M \mid K \mid \{m\}_n.K) \mid \{m'\}_n.\{\bar{m}\}_n.0) \simeq_{\text{Ct}}^S (\nu m)((\nu \tilde{c})(N \mid L \mid \{m\}_n.L) \mid \{m'\}_n.\{\bar{m}\}_n.0)$. Therefore we get $(\nu \tilde{b})(M \mid K \mid \{m'\}_n.K) R (\nu \tilde{c})(N \mid L \mid \{m'\}_n.L)$.

If $I\alpha \neq \{m'\}_n$, since $m' \notin \text{fn}(K, L, M, N) \cup \{\tilde{b}\} \cup \{\tilde{c}\} \cup \{m\}$, we have $(\nu \tilde{b})(M \mid \{m'\}_n.K) \xrightarrow{I\alpha, S} (\nu \tilde{b})(M' \mid \{m'\}_n.K)$ and $(\nu \tilde{c})(N \mid \{m'\}_n.L) \xrightarrow{I\alpha, S} (\nu \tilde{c})(N' \mid \{m'\}_n.L)$. Since $(\nu m)((\nu \tilde{b})(M \mid \{m\}_n.K) \mid \{m'\}_n.\{\bar{m}\}_n.0) \xrightarrow{I\alpha, S} (\nu m)((\nu \tilde{b})(M' \mid \{m\}_n.K) \mid \{m'\}_n.\{\bar{m}\}_n.0)$

$\{m'\}_n.\{\bar{m}\}_n.0$ and $(\nu m)((\nu \tilde{b})(M'!\{m\}_n.K)!\{m'\}_n.\{\bar{m}\}_n.0) \simeq_{\text{ct}}^S (\nu m)((\nu \tilde{c})(N'!\{m\}_n.L)!\{m'\}_n.\{\bar{m}\}_n.0)$, we have $(\nu m)((\nu \tilde{c})(N'!\{m\}_n.L)!\{m'\}_n.\{\bar{m}\}_n.0) \xrightarrow{I\alpha, S} (\nu m)((\nu \tilde{c})(N''!\{m\}_n.L)!\{m'\}_n.\{\bar{m}\}_n.0)$ and $(\nu m)((\nu \tilde{b})(M''!\{m\}_n.K)!\{m'\}_n.\{\bar{m}\}_n.0) \simeq_{\text{ct}}^S (\nu m)((\nu \tilde{c})(N''!\{m\}_n.L)!\{m'\}_n.\{\bar{m}\}_n.0)$. Therefore $(\nu \tilde{b})(M''!\{m\}_n.K) R (\nu \tilde{c})(N''!\{m\}_n.L)$. \square

Lemma 1. For arbitrary $M, N \in \text{ITPr}^c$, $M \simeq_{\text{ct}}^S N \Rightarrow M \simeq_{\text{tr}}^S N$.

Proof. We prove that $R = \{(M, N) : M \simeq_{\text{ct}}^S N\}$ is a \simeq_{tr}^S bisimulation. The only nontrivial case is $M \xrightarrow{\{(vm)\bar{a}(\{\bar{m}\}_n.0)\}_{i,S}} M'$, so we consider only this case.

Suppose $M \xrightarrow{\{(vm)\bar{a}(\{\bar{m}\}_n.0)\}_{i,S}} M'$. Since $M \simeq_{\text{ct}}^S N$, there exists N' such that $N \xrightarrow{\{(vm)\bar{a}(\{\bar{m}\}_n.0)\}_{i,S}} N'$ and for every $C(U)$ with $m \notin \text{fn}(C(U))$, $(\nu m)(M'|C(\{\bar{m}\}_n.0)) \simeq_{\text{ct}}^S (\nu m)(N'|C(\{\bar{m}\}_n.0))$. In order to close the bisimulation we have to show that $M' \simeq_{\text{ct}}^S N'$.

By Lemma A.1, there exist $\tilde{b}, \tilde{c}, K, L, M'', N''$ s.t. $M' \simeq_{\text{ct}}^S (\nu \tilde{b})(M''!\{m\}_n.K)$ and $N' \simeq_{\text{ct}}^S (\nu \tilde{c})(N''!\{m\}_n.L)$ with $m \notin \text{fn}(K, L, M'', N'') \cup \{\tilde{b}\} \cup \{\tilde{c}\}$. Suppose m' is a fresh name. Now changing m for m' does not affect the equivalence among processes, i.e., we have $(\nu b)(M''!\{m\}_n.K) \simeq_{\text{ct}}^S (\nu \tilde{c})(N''!\{m\}_n.L)$ iff $(\nu b)(M''!\{m'\}_n.K) \simeq_{\text{ct}}^S (\nu \tilde{c})(N''!\{m'\}_n.L)$. Therefore if we can prove the latter equivalence we get $M' \simeq_{\text{ct}}^S N'$.

Since for every $C(U)$ with $m \notin \text{fn}(C(U))$, $(\nu m)(M'|C(\{\bar{m}\}_n.0)) \simeq_{\text{ct}}^S (\nu m)(N'|C(\{\bar{m}\}_n.0))$. Let $C(U) = !\{m'\}_n.U$, then $(\nu m)((\nu \tilde{b})(M''!\{m\}_n.K)!\{m'\}_n.\{\bar{m}\}_n.0) \simeq_{\text{ct}}^S (\nu m)((\nu \tilde{c})(N''!\{m\}_n.L)!\{m'\}_n.\{\bar{m}\}_n.0)$. By Lemma A.2, we have $(\nu \tilde{b})(M''!\{m'\}_n.K) \simeq_{\text{ct}}^S (\nu \tilde{c})(N''!\{m'\}_n.L)$, this concludes the analysis. \square

Appendix B. Proof of Proposition 6

Definition B.1. A relation $R \subseteq \text{ITPr}^c \times \text{ITPr}^c$ is an indexed triggered bisimulation up to \simeq_{tr}^S , if there is a symmetric relation R :

- (1) whenever $M R N$ and $M \xrightarrow{\varepsilon, S} M'$, there exists N' such that $N \xrightarrow{\varepsilon, S} N'$ and $M' \simeq_{\text{tr}}^S R \simeq_{\text{tr}}^S N'$;
- (2) whenever $M R N$ and $M \xrightarrow{I\alpha, S} M'$, there exists N' such that $N \xrightarrow{I\alpha, S} N'$ and $M' \simeq_{\text{tr}}^S R \simeq_{\text{tr}}^S N'$, where $I\alpha \neq \{\tau\}_{i,i}$ for any $i \in S$, $I\alpha$ is not an indexed higher order action;
- (3) whenever $M R N$ and $M \xrightarrow{\{a(\{\bar{m}\}_n.0)\}_{i,S}} M'$, where m is a fresh name, there exists N' such that $N \xrightarrow{\{a(\{\bar{m}\}_n.0)\}_{i,S}} N'$ and $M' \simeq_{\text{tr}}^S R \simeq_{\text{tr}}^S N'$;
- (4) whenever $M R N$ and $M \xrightarrow{\{(vm)\bar{a}(\{\bar{m}\}_n.0)\}_{i,S}} M'$, there exists N' such that $N \xrightarrow{\{(vm)\bar{a}(\{\bar{m}\}_n.0)\}_{i,S}} N'$ and $M' \simeq_{\text{tr}}^S R \simeq_{\text{tr}}^S N'$.

Lemma B.1. If R is an indexed triggered bisimulation up to \simeq_{tr}^S , then $R \subseteq \simeq_{\text{tr}}^S$.

Proof. The same argument of the analogous result for CCS bisimilarity in [13]: use a diagram-chasing argument to show that $\simeq_{\text{tr}}^S R \simeq_{\text{tr}}^S$ is an indexed triggered bisimulation. \square

Proposition 6. For any $M, N \in \text{IPr}^c$, $M \simeq_{\text{Nr}}^S N \Rightarrow \text{Tr}^n[M] \simeq_{\text{tr}}^{S \cup \{n\}} \text{Tr}^n[N]$, where $n \notin \text{ix}(M, N)$.

Proof. We show that $R = \{(\text{Tr}^n[M], \text{Tr}^n[N]) : M \simeq_{\text{Nr}}^S N\}$ is a $\simeq_{\text{tr}}^{S \cup \{n\}}$ -bisimulation up to $\simeq_{\text{tr}}^{S \cup \{n\}}$.

The only nontrivial case is to show how indexed higher order output actions of $\text{Tr}^n[M]$ are matched by $\text{Tr}^n[N]$.

Suppose $\text{Tr}^n[M] \xrightarrow{\{(vm)\bar{a}(\{\bar{m}\}_n.0)\}_{i,S \cup \{n\}}} M''$ for $m \notin \text{fn}(M, N)$. By Lemma 2, M', K , and \tilde{b} exist s.t. $M \xrightarrow{\{(v\tilde{b})\bar{a}(K)\}_{i,S}} M'$ and $M'' \simeq_{\text{tr}}^{S \cup \{n\}} \text{Tr}^n[(\nu \tilde{b})(M'!\{m\}_n.K)]$.

Since $M \simeq_{\text{Nr}}^S N$, there exist N', L , and \tilde{b} exist such that $N \xrightarrow{\{(v\tilde{b})\bar{a}(L)\}_{i,S}} N'$ and $(\nu \tilde{b})(M'!\{m\}_n.K) \simeq_{\text{Nr}}^S (\nu \tilde{c})(N'!\{m\}_n.L)$.

Further, by Lemma 2, $\text{Tr}^n[N] \xrightarrow{\{(vm)\bar{a}(\{\bar{m}\}_n.0)\}_{i,S \cup \{n\}}} N'' \simeq_{\text{tr}}^{S \cup \{n\}} \text{Tr}^n[(\nu \tilde{c})(N'!\{m\}_n.L)]$.

Summarizing, we have $\text{Tr}^n[M] \xrightarrow{\{(vm)\bar{a}(\{\bar{m}\}_n.0)\}_{i,S \cup \{n\}}} M'' \simeq_{\text{tr}}^{S \cup \{n\}} \text{Tr}^n[(\nu \tilde{b})(M'!\{m\}_n.K)] R \text{Tr}^n[(\nu \tilde{c})(N'!\{m\}_n.L)] \simeq_{\text{tr}}^{S \cup \{n\}} N'' \xleftarrow{\{(vm)\bar{a}(\{\bar{m}\}_n.0)\}_{i,S \cup \{n\}}} \text{Tr}^n[N]$.

Hence R is a $\simeq_{\text{tr}}^{S \cup \{n\}}$ -bisimulation up to $\simeq_{\text{tr}}^{S \cup \{n\}}$ and $R \subseteq \simeq_{\text{tr}}^{S \cup \{n\}}$. \square

Appendix C. Proof of Lemma 4

Definition C.1. A relation $R \subseteq \text{IPr}^c \times \text{IPr}^c$ is an indexed context bisimulation up to \simeq_{ct}^S , if there is a symmetric relation R :

- (1) whenever $M R N$ and $M \xrightarrow{\varepsilon, S} M'$, there exists N' such that $N \xrightarrow{\varepsilon, S} N'$ and $M' \simeq_{\text{ct}}^S R \simeq_{\text{ct}}^S N'$;
- (2) whenever $M R N$ and $M \xrightarrow{I\alpha, S} M'$, there exists N' such that $N \xrightarrow{I\alpha, S} N'$ and $M' \simeq_{\text{ct}}^S R \simeq_{\text{ct}}^S N'$, where $I\alpha \neq \{\tau\}_{i,i}$ for any $i \in S$, $I\alpha$ is not an indexed higher order action;
- (3) whenever $M R N$ and $M \xrightarrow{\{a(K)\}_{i,S}} M'$, there exists N' such that $N \xrightarrow{\{a(K)\}_{i,S}} N'$ and $M' \simeq_{\text{ct}}^S R \simeq_{\text{ct}}^S N'$;
- (4) whenever $M R N$ and $M \xrightarrow{\{(v\tilde{b})\bar{a}(K)\}_{i,S}} M'$, there exists N' such that $N \xrightarrow{\{(v\tilde{c})\bar{a}(L)\}_{i,S}} N'$ and for any indexed process $C(U)$ with $\text{fn}(C(U)) \cap \{\tilde{b}, \tilde{c}\} = \emptyset$, $(\nu \tilde{b})(M'|C(K)) \simeq_{\text{ct}}^S R \simeq_{\text{ct}}^S (\nu \tilde{c})(N'|C(L))$.

Lemma C.1. If R is an indexed context bisimulation up to \simeq_{Ct}^S , then $R \subseteq \simeq_{\text{Ct}}^S$.

Proof. Similar to the analogous result for CCS bisimilarity in [13], use a diagram-chasing argument to show that $\simeq_{\text{Ct}}^S R \simeq_{\text{Ct}}^S$ is an indexed context bisimulation. \square

Lemma 4. For any $M, N \in \text{IPr}^c$, $\text{Tr}^n[M] \xrightarrow{\text{Tr}}^{\text{SU}\{n\}} \text{Tr}^n[N] \Rightarrow M \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} N$, where $n \notin \text{ix}(M, N)$.

Proof. We show that $R = \{(M, N) : \text{Tr}^n[M] \xrightarrow{\text{Tr}}^{\text{SU}\{n\}} \text{Tr}^n[N]\}$ is a $\simeq_{\text{Ct}}^{\text{SU}\{n\}}$ -bisimulation up to $\simeq_{\text{Ct}}^{\text{SU}\{n\}}$.

Let $(M, N) \in R$.

We first discuss the case of indexed higher order input action.

Suppose $M \xrightarrow{\{a(K)\}_i, \text{SU}\{n\}} M'$. We have to find N', N'' and M'' such that $N \xrightarrow{\{a(K)\}_i, \text{SU}\{n\}} N'$ and $M' \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} M'' R N'' \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} N'$.

Let $M \xrightarrow{\{a(K)\}_i, \text{SU}\{n\}} M'$. Using Lemma 3, there exists M_1 such that if $m \notin \text{fn}(M, N)$, then $\text{Tr}^n[M] \xrightarrow{\{a(\{\bar{m}\}_n, 0)\}_i, \text{SU}\{n\}} M_1$ and $(\nu m)(M_1 | \{m\}_n.K) \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} (\nu m)(M_1 | \{m\}_n.\text{Tr}^n[K]) \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} \text{Tr}^n[M']$.

From $\text{Tr}^n[M] \xrightarrow{\text{Tr}}^{\text{SU}\{n\}} \text{Tr}^n[N]$ and $\text{Tr}^n[M] \xrightarrow{\{a(\{\bar{m}\}_n, 0)\}_i, \text{SU}\{n\}} M_1$, we get that there exists N_1 such that $\text{Tr}^n[N] \xrightarrow{\{a(\{\bar{m}\}_n, 0)\}_i, \text{SU}\{n\}} N_1$ and $N_1 \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} M_1$. By the congruence properties of $\simeq_{\text{Tr}}^{\text{SU}\{n\}}$, we also have $(\nu m)(M_1 | \{m\}_n.\text{Tr}^n[K]) \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} (\nu m)(N_1 | \{m\}_n.\text{Tr}^n[K])$.

Reversing the reasoning, we find N' such that $N \xrightarrow{\{a(K)\}_i, \text{SU}\{n\}} N' \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} \text{Tr}^n[N'] \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} (\nu m)(N_1 | \{m\}_n.\text{Tr}^n[K])$.

Now, we have $M \xrightarrow{\{a(K)\}_i, \text{SU}\{n\}} M' \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} (\nu m)(M_1 | \{m\}_n.\text{Tr}^n[K]) \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} (\nu m)(N_1 | \{m\}_n.\text{Tr}^n[K]) \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} N' \xleftarrow{\{a(K)\}_i, \text{SU}\{n\}} N$. Let $M'' = (\nu m)(M_1 | \{m\}_n.\text{Tr}^n[K])$ and $N'' = (\nu m)(N_1 | \{m\}_n.\text{Tr}^n[K])$. It is not difficult to see that M'' and N'' are all indexed triggered processes, hence $M'' \xrightarrow{\text{Tr}}^{\text{SU}\{n\}} \text{Tr}^n[M'']$ and $N'' \xrightarrow{\text{Tr}}^{\text{SU}\{n\}} \text{Tr}^n[N'']$, we have $M' \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} M'' R N'' \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} N'$, this concludes the analysis.

Now we discuss the case of indexed higher order output action.

Suppose $M \xrightarrow{\{(\nu \tilde{b})\bar{a}(K)\}_i, \text{SU}\{n\}} M'$. We have to find N', N'' and M'' such that $N \xrightarrow{\{(\nu \tilde{c})\bar{a}(L)\}_i, \text{SU}\{n\}} N'$ and for any indexed process $C(U)$ with $\text{fn}(C(U)) \cap \{\tilde{b}, \tilde{c}, m\} = \emptyset$, $(\nu \tilde{b})(M' | C(K)) \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} M'' R N'' \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} (\nu \tilde{c})(N' | C(L))$.

Let $M \xrightarrow{\{(\nu \tilde{b})\bar{a}(K)\}_i, \text{SU}\{n\}} M'$. We have $\text{Tr}^n[M] \xrightarrow{\{(\nu m)\bar{a}(\{\bar{m}\}_n, 0)\}_i, \text{SU}\{n\}} M_1$ and for any indexed process $C(U)$ with $\text{fn}(C(U)) \cap \{\tilde{b}, \tilde{c}, m\} = \emptyset$, $(\nu m)(M_1 | \text{Tr}^n[C(\{\bar{m}\}_n, 0)]) \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} \text{Tr}^n[(\nu \tilde{b})(M' | C(K))]$.

Since $\text{Tr}^n[M] \xrightarrow{\text{Tr}}^{\text{SU}\{n\}} \text{Tr}^n[N]$, $\text{Tr}^n[N] \xrightarrow{\{(\nu m)\bar{a}(\{\bar{m}\}_n, 0)\}_i, \text{SU}\{n\}} N_1$ and $M_1 \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} N_1$. By the congruence properties of $\simeq_{\text{Tr}}^{\text{SU}\{n\}}$, we have $(\nu m)(N_1 | \text{Tr}^n[C(\{\bar{m}\}_n, 0)]) \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} (\nu m)(M_1 | \text{Tr}^n[C(\{\bar{m}\}_n, 0)])$.

By Lemma 3, $N \xrightarrow{\{(\nu \tilde{c})\bar{a}(L)\}_i, \text{SU}\{n\}} N'$ and $(\nu m)(N_1 | \text{Tr}^n[C(\{\bar{m}\}_n, 0)]) \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} \text{Tr}^n[(\nu \tilde{c})(N' | C(L))]$.

Hence $M \xrightarrow{\{(\nu \tilde{b})\bar{a}(K)\}_i, \text{SU}\{n\}} M', N \xrightarrow{\{(\nu \tilde{c})\bar{a}(L)\}_i, \text{SU}\{n\}} N'$ and $(\nu \tilde{b})(M' | C(K)) \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} \text{Tr}^n[(\nu \tilde{b})(M' | C(K))] \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} (\nu m)(M_1 | \text{Tr}^n[C(\{\bar{m}\}_n, 0)]) \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} (\nu m)(N_1 | \text{Tr}^n[C(\{\bar{m}\}_n, 0)]) \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} \text{Tr}^n[(\nu \tilde{c})(N' | C(L))] \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} (\nu \tilde{c})(N' | C(L))$. Let $M'' = (\nu m)(M_1 | \text{Tr}^n[C(\{\bar{m}\}_n, 0)])$ and $N'' = (\nu m)(N_1 | \text{Tr}^n[C(\{\bar{m}\}_n, 0)])$. Since it is easy to see that M'' and N'' are all indexed triggered processes, we have $M'' \xrightarrow{\text{Tr}}^{\text{SU}\{n\}} \text{Tr}^n[M'']$ and $N'' \xrightarrow{\text{Tr}}^{\text{SU}\{n\}} \text{Tr}^n[N'']$, hence $(\nu \tilde{b})(M' | C(K)) \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} M'' R N'' \xrightarrow{\text{Ct}}^{\text{SU}\{n\}} (\nu \tilde{c})(N' | C(L))$, this concludes the analysis. \square

Appendix D. Proof of Proposition 9

Proposition 9. For arbitrary $M, N \in \text{IPr}^c$, $M \xrightarrow{\text{Ct}}^S N \Leftrightarrow M \xrightarrow{\text{Ba}}^S N \Leftrightarrow M \xrightarrow{\text{Rd}}^S N \Leftrightarrow M \xrightarrow{\text{Nr}}^S N$.

Proof. (1) $M \xrightarrow{\text{Ct}}^S N \Rightarrow M \xrightarrow{\text{Ba}}^S N$: Let $R = \{(M, N) : M \xrightarrow{\text{Ct}}^S N, \text{ where } M, N \text{ are indexed processes}\}$. To prove $M \xrightarrow{\text{Ct}}^S N \Rightarrow M \xrightarrow{\text{Ba}}^S N$, it is enough to prove that $R \subseteq \simeq_{\text{Ba}}^S$.

(a) Firstly, suppose $(M, N) \in R$, we want to prove that for any indexed process C , if $C|M \xrightarrow{\{\tau\}_{i,j}, S} K$, then $C|N \xrightarrow{\{\tau\}_{i,j}, S} L$ and $(K, L) \in R$.

We only discuss the following cases, other cases are similar or direct.

Case 1. $M \xrightarrow{\{\tau\}_{i,j}, S} M'$. Hence we have $C|M \xrightarrow{\{\tau\}_{i,j}, S} C|M'$. Since $M \xrightarrow{\text{Ct}}^S N$, there is N' such that $N \xrightarrow{\{\tau\}_{i,j}, S} N'$. Therefore $C|N \xrightarrow{\{\tau\}_{i,j}, S} C|N'$ and $M' \xrightarrow{\text{Ct}}^S N'$. By the congruence of \simeq_{Ct}^S (Proposition 2), $C|M' \xrightarrow{\text{Ct}}^S C|N'$. Hence $(C|M', C|N') \in R$.

Case 2. $M \xrightarrow{\{(\nu \tilde{b})\bar{a}(K)\}_i, S} M'$. Let $C \equiv \{a(U)\}_j.C'(U)$. Hence we have $C|M \xrightarrow{\{\tau\}_{i,j}, S} (\nu \tilde{b})(C' | K | M')$. Since $M \xrightarrow{\text{Ct}}^S N$, there is $(\nu \tilde{c})\bar{a}(L)$ such that $N \xrightarrow{\{(\nu \tilde{c})\bar{a}(L)\}_i, S} N'$. Therefore $C|N \xrightarrow{\{\tau\}_{i,j}, S} (\nu \tilde{c})(C' | L | N')$ and $(\nu \tilde{b})(C' | K | M') \xrightarrow{\text{Ct}}^S (\nu \tilde{c})(C' | L | N')$. Hence $((\nu \tilde{b})(C' | K | M'), (\nu \tilde{c})(C' | L | N')) \in R$.

(b) Suppose $M \Downarrow_a^S$. There exists M' , such that $M \xRightarrow{e, S} M' \Downarrow_a$. Since $M \xrightarrow{\text{Ct}}^S N$, it is easy to see that there exists N' , such that $N \xRightarrow{e, S} N' \Downarrow_a$. Therefore $N \Downarrow_a^S$.

(2) $M \simeq_{Ba}^S N \Rightarrow M \simeq_{Rd}^S N$: Let $R = \{(M, N) : M \simeq_{Ba}^S N, \text{ where } M, N \text{ are indexed processes}\}$.

By the definition, it is trivial to prove that $R \subseteq \simeq_{Rd}^S$.

(3) $M \simeq_{Rd}^S N \Rightarrow M \simeq_{Nr}^S N$: Let $R = \{(M, N) : M \simeq_{Rd}^S N, \text{ where } M, N \text{ are indexed processes}\}$. To prove $M \simeq_{Rd}^S N \Rightarrow M \simeq_{Nr}^S N$, it is enough to prove that $R \subseteq \simeq_{Nr}^S$.

Suppose $(M, N) \in R$, we need to prove that if $M \xrightarrow{I\alpha, S} M'$, then

(a) $N \xrightarrow{I\alpha, S} N'$ and $(M', N') \in R$, where $I\alpha$ is not an indexed higher order action;

(b) $N \xrightarrow{\{a(\{\bar{m}\}_n, 0)\}_i, S} N'$ and $(M', N') \in R$, where $I\alpha = \{a(\{\bar{m}\}_n, 0)\}_i$.

(c) $N \xrightarrow{\{(v\tilde{c})\bar{a}(L)\}_i, S} N'$ with $I\alpha = \{(v\tilde{b})\bar{a}(K)\}_i$, and $((v\tilde{b})(M'! \{m\}_n, K), (v\tilde{c})(N'! \{m\}_n, L)) \in R$ with a fresh name m and a fresh index n .

We only discuss the following cases, other cases are similar or direct.

Case 1. Let $M \xrightarrow{\{\tau\}_{ij}, S} M'$, then for testing process $\{0\}_1$, we have $\{0\}_1 | M \simeq_{Rd}^S \{0\}_1 | N$. Since $\{0\}_1 | M \xrightarrow{\{\tau\}_{ij}, S} \{0\}_1 | M'$, we have $\{0\}_1 | N \xrightarrow{\{\tau\}_{ij}, S} \{0\}_1 | N'$, and $\{0\}_1 | M' \simeq_{Rd}^S \{0\}_1 | N'$. Hence we have $N \xrightarrow{\{\tau\}_{ij}, S} N'$ with $(M', N') \in R$.

Case 2. Let $M \xrightarrow{\{(v\tilde{b})\bar{a}(K)\}_i, S} M'$, then for testing process $\{a(U)\}_j.!\{m\}_n.U$, we have $\{a(U)\}_j.!\{m\}_n.U | M \simeq_{Rd}^S \{a(U)\}_j.!\{m\}_n.U | N$. Since we have $\{a(U)\}_j.!\{m\}_n.U | M \xrightarrow{\{\tau\}_{ij}, S} (v\tilde{b})(M'! \{m\}_n, K)$, there is $\{a(U)\}_j.!\{m\}_n.U | N \xrightarrow{\{\tau\}_{ij}, S} (v\tilde{c})(N'! \{m\}_n, L)$ and $(v\tilde{b})(M'! \{m\}_n, K) \simeq_{Rd}^S (v\tilde{c})(N'! \{m\}_n, L)$. By the construction of $\{a(U)\}_j.!\{m\}_n.U$, we have $N \xrightarrow{\{(v\tilde{c})\bar{a}(L)\}_i, S} N'$ with $((v\tilde{b})(M'! \{m\}_n, K), (v\tilde{c})(N'! \{m\}_n, L)) \in R$.

(4) $M \simeq_{Nr}^S N \Rightarrow M \simeq_{Ct}^S N$: By Proposition 8. \square

Appendix E. Proof of Lemma 7

In the following, we abbreviate $P\{n_1/a_1, \dots, n_k/a_k\}$ (or $\{P\}_{i_0}\{\{n_1\}_{i_1}/\{a_1\}_{i_0}, \dots, \{n_k\}_{i_k}/\{a_k\}_{i_0}\}$) as $P\langle n_1, \dots, n_k \rangle$ (or $\{P\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle$), where $P\{n_1/a_1, \dots, n_k/a_k\}$ denotes the componentwise and simultaneous substitution of names n_1, \dots, n_k with names a_1, \dots, a_k in the scope of P ; $\{P\}_{i_0}\{\{n_1\}_{i_1}/\{a_1\}_{i_0}, \dots, \{n_k\}_{i_k}/\{a_k\}_{i_0}\}$ denotes the componentwise and simultaneous substitution of indexed names $\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}$ with indexed names $\{a_1\}_{i_0}, \dots, \{a_k\}_{i_0}$ in the scope of $\{P\}_{i_0}$, and as we defined before, where $\{P\}_{i_0}$ denotes indexed process with the same given index i_0 on every prefix in the scope of P .

If a prefix n occurs in process $n.A$, then we say that n is in a first order input subject position of $n.A$, and similarly if a prefix \bar{n} occurs in process $\bar{n}.A$, then we say that n is in a first order output subject position of $\bar{n}.A$. We use $fisp(P)$ to denote the set of names that occur free in process P only in first order input subject positions, and $fosp(P)$ to denote the set of names that occur free in process P only in first order output subject positions.

Lemma 7. For any $P, Q \in Pr^c$, $P \sim_{Nr} Q \Rightarrow P \sim_{Nr}^i Q$.

Proof. Let $R = \{(\{P\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle, \{Q\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle) : P\langle n_1, \dots, n_k \rangle \sim_{Nr} Q\langle n_1, \dots, n_k \rangle, \text{ where for any } m = 1, \dots, k, n_m \text{ is a fresh name and either (1) } n_m \in fisp(P\langle n_1, \dots, n_k \rangle) \text{ and } n_m \in fisp(Q\langle n_1, \dots, n_k \rangle) \text{ or (2) } n_m \in fosp(P\langle n_1, \dots, n_k \rangle) \text{ and } n_m \in fosp(Q\langle n_1, \dots, n_k \rangle))\}$.

Suppose $\{P\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle R \{Q\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle$ and $\{P\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle \xrightarrow{I\alpha} M'$.

Case (1): $I\alpha$ is in the form of $\{l\}_i$ or $\{\bar{l}\}_i$, w.l.o.g., we assume that $I\alpha = \{l\}_i$.

If $\{P\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle \xrightarrow{\{l\}_i} \{P'\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle$, there are two sub-cases: (a) $l = n_j \in \{n_1, \dots, n_k\}$, and $\{P\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle \xrightarrow{\{n_j\}_{i_j}} \{P'\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle$; (b) $l \notin \{n_1, \dots, n_k\}$, and $\{P\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle \xrightarrow{\{l\}_{i_0}} \{P'\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle$.

We have the corresponding transition $P\langle n_1, \dots, n_k \rangle \xrightarrow{l} P'\langle n_1, \dots, n_k \rangle$. Since $P\langle n_1, \dots, n_k \rangle \sim_{Nr} Q\langle n_1, \dots, n_k \rangle$, $Q\langle n_1, \dots, n_k \rangle \xrightarrow{l} Q'\langle n_1, \dots, n_k \rangle$ and $P'\langle n_1, \dots, n_k \rangle \sim_{Nr} Q'\langle n_1, \dots, n_k \rangle$.

In sub-case (a), since n_j is a fresh name and is indexed by i_j , we have $\{Q\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle \xrightarrow{\{n_j\}_{i_j}} \{Q'\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle$ and $\{P'\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle R \{Q'\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle$.

In sub-case (b), since $l \notin \{n_1, \dots, n_k\}$, l is indexed by i_0 . Therefore we have $\{Q\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle \xrightarrow{\{l\}_{i_0}} \{Q'\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle$ and $\{P'\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle R \{Q'\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle$.

Case (2): $I\alpha$ is in the form of $\{a(\{\bar{n}\}_i, 0)\}_j$. Since any higher order prefix in $\{P\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle$ is indexed by i_0 , we have $j = i_0$.

Let $\{P\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k} \rangle \xrightarrow{\{a(\{\bar{n}\}_i, 0)\}_{i_0}} \{P'\}_{i_0}\langle \{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}, \{n\}_i \rangle$, where n is a fresh name and $n \in fosp(P'\langle n_1, \dots, n_k, n \rangle)$, then we have $P\langle n_1, \dots, n_k \rangle \xrightarrow{a(\bar{n}, 0)} P'\langle n_1, \dots, n_k, n \rangle$.

Since $P\langle n_1, \dots, n_k \rangle \sim_{Nr} Q\langle n_1, \dots, n_k \rangle$, $Q\langle n_1, \dots, n_k \rangle \xrightarrow{a(\bar{n}.0)} Q'\langle n_1, \dots, n_k, n \rangle$ and $P'\langle n_1, \dots, n_k, n \rangle \sim_{Nr} Q'\langle n_1, \dots, n_k, n \rangle$, where $n \in \text{fosp}(P\langle n_1, \dots, n_k, n \rangle)$ and $n \in \text{fosp}(Q'\langle n_1, \dots, n_k, n \rangle)$.

Since any higher order prefix in $\{Q\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$ is indexed by i_0 , we have $\{Q\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle \xrightarrow{\{a(\bar{n}.0)\}_{i_0}} \{Q'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}, \{n\}_i\rangle$ and $\{P'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}, \{n\}_i\rangle R \{Q'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}, \{n\}_i\rangle$.

Case (3): $I\alpha$ is in the form of $\{(\nu\tilde{e})\tilde{a}\langle\{E\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle\rangle_j\}$. Since any higher order prefix in $\{P\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$ is indexed by i_0 , we have $j = i_0$.

Let $\{P\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle \xrightarrow{\{(\nu\tilde{e})\tilde{a}\langle\{E\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle\rangle_{i_0}\}} \{P'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$, we have $P\langle n_1, \dots, n_k \rangle \xrightarrow{(\nu\tilde{e})\tilde{a}\langle E(n_1, \dots, n_k) \rangle} P'\langle n_1, \dots, n_k \rangle$.

Since $P\langle n_1, \dots, n_k \rangle \sim_{Nr} Q\langle n_1, \dots, n_k \rangle$, we have $Q\langle n_1, \dots, n_k \rangle \xrightarrow{(\nu\tilde{f})\tilde{a}\langle F(n_1, \dots, n_k) \rangle} Q'\langle n_1, \dots, n_k \rangle$ and $(\nu\tilde{e})(P'\langle n_1, \dots, n_k \rangle!n.E\langle n_1, \dots, n_k \rangle) = S'\langle n_1, \dots, n_k, n \rangle \sim_{Nr} T'\langle n_1, \dots, n_k, n \rangle = (\nu\tilde{f})(Q'\langle n_1, \dots, n_k \rangle!n.F\langle n_1, \dots, n_k \rangle)$, where n is a fresh name, $n \in \text{fisp}((\nu\tilde{e})(P'\langle n_1, \dots, n_k \rangle!n.E\langle n_1, \dots, n_k \rangle))$ and $n \in \text{fisp}((\nu\tilde{f})(Q'\langle n_1, \dots, n_k \rangle!n.F\langle n_1, \dots, n_k \rangle))$.

Since any higher order prefix in $\{Q\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$ is indexed by i_0 , we have $\{Q\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle \xrightarrow{\{(\nu\tilde{f})\tilde{a}\langle\{F\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle\rangle_{i_0}\}} \{Q'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$ and $(\nu\tilde{e})(\{P'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle!n_j.E\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle) = \{S'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}, \{n_j\}\rangle R \{T'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}, \{n_j\}\rangle = (\nu\tilde{f})(\{Q'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle!n_j.F\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle)$.

Case (4): $I\alpha$ is in the form of $\{\tau\}_{i,j}$.

Suppose $\{P\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle \xrightarrow{\{\tau\}_{i,j}} \{P'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$. Since for any $m = 1, \dots, k$, either $n_m \in \text{fisp}(P\langle n_1, \dots, n_k \rangle)$ or $n_m \in \text{fosp}(P\langle n_1, \dots, n_k \rangle)$, we have $i = j = i_0$ and $P\langle n_1, \dots, n_k \rangle \xrightarrow{\tau} P'\langle n_1, \dots, n_k \rangle$.

Since $P\langle n_1, \dots, n_k \rangle \sim_{Nr} Q\langle n_1, \dots, n_k \rangle$, $Q\langle n_1, \dots, n_k \rangle \xrightarrow{\tau} Q'\langle n_1, \dots, n_k \rangle$ and $P'\langle n_1, \dots, n_k \rangle \sim_{Nr} Q'\langle n_1, \dots, n_k \rangle$.

Hence $\{Q\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle \xrightarrow{\{\tau\}_{i,j}} \{Q'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$. Furthermore, since for any $m = 1, \dots, k$, either $n_m \in \text{fisp}(Q\langle n_1, \dots, n_k \rangle)$ or $n_m \in \text{fosp}(Q\langle n_1, \dots, n_k \rangle)$, we have $i = j = i_0$. Hence $\{Q\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle \xrightarrow{\{\tau\}_{i_0, i_0}} \{Q'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$ and $\{P'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle R \{Q'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$. \square

Appendix F. Proof of Lemma 8

Lemma 8. For any $P, Q \in \text{Pr}^c$, $P \approx_{Nr} Q \Rightarrow P \approx_{Nr}^i Q$.

Proof. Let $R = \{(\{P\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle, \{Q\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle) : P\langle n_1, \dots, n_k \rangle \approx_{Nr} Q\langle n_1, \dots, n_k \rangle, \text{ where for any } m = 1, \dots, k, n_m \text{ is a fresh name and either (1) } n_m \in \text{fisp}(P\langle n_1, \dots, n_k \rangle) \text{ and } n_m \in \text{fisp}(Q\langle n_1, \dots, n_k \rangle) \text{ or (2) } n_m \in \text{fosp}(P\langle n_1, \dots, n_k \rangle) \text{ and } n_m \in \text{fosp}(Q\langle n_1, \dots, n_k \rangle))\}$.

Suppose $\{P\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle R \{Q\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$. We only discuss the case $\{P\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle \xrightarrow{\{\tau\}_{i,j}, I} M'$, other cases are similar to the argument in the proof of Lemma 9.

Let $\{P\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle \xrightarrow{\{\tau\}_{i,j}, I} \{P'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$. Since for any $m = 1, \dots, k$, either $n_m \in \text{fisp}(P\langle n_1, \dots, n_k \rangle)$ or $n_m \in \text{fosp}(P\langle n_1, \dots, n_k \rangle)$, we have $i = j = i_0$ and $P\langle n_1, \dots, n_k \rangle \xrightarrow{\tau} P'\langle n_1, \dots, n_k \rangle$.

Since $P\langle n_1, \dots, n_k \rangle \approx_{Nr} Q\langle n_1, \dots, n_k \rangle$, $Q\langle n_1, \dots, n_k \rangle \xrightarrow{\tau} Q'\langle n_1, \dots, n_k \rangle$ and $P'\langle n_1, \dots, n_k \rangle \approx_{Nr} Q'\langle n_1, \dots, n_k \rangle$.

We have the corresponding indexed transition $\{Q\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle \xrightarrow{\{\tau\}_{p_1, q_1}} \dots \xrightarrow{\{\tau\}_{p_r, q_r}} \{Q'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$. Furthermore, since for any $m = 1, \dots, k$, either $n_m \in \text{fisp}(Q\langle n_1, \dots, n_k \rangle)$ or $n_m \in \text{fosp}(Q\langle n_1, \dots, n_k \rangle)$, we have $p_1 = q_1 = \dots = p_r = q_r = i_0$. Hence $\{Q\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle \xrightarrow{\{\tau\}_{i_0, i_0}} \dots \xrightarrow{\{\tau\}_{i_0, i_0}} \{Q'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$ and $\{P'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle R \{Q'\}_{i_0}\langle\{n_1\}_{i_1}, \dots, \{n_k\}_{i_k}\rangle$. \square

Appendix G. Proof of Lemma 10

Lemma 10. $P \approx_{nor} Q \Rightarrow P \approx_{Nr} Q$.

Proof. Let $R = \{((\nu\tilde{b})(P!m_1.E_1 \dots !m_n.E_n), (\nu\tilde{c})(Q!m_1.F_1 \dots !m_n.F_n)) : (\nu\tilde{b})(P!m_1.E_1 \dots !m_n.E_n) \approx_{nor} (\nu\tilde{c})(Q!m_1.F_1 \dots !m_n.F_n) \text{ for any } k_1, \dots, k_n \in N \text{ with fresh names } m_1, \dots, m_n\}$. It is enough to prove that $R \subseteq \approx_{Nr}$.

We want to prove that

(1) If $(\nu\tilde{b})(P!m_1.E_1 \dots !m_n.E_n) R (\nu\tilde{c})(Q!m_1.F_1 \dots !m_n.F_n)$ and $(\nu\tilde{b})(P!m_1.E_1 \dots !m_n.E_n) \xrightarrow{\tau} S$, then there exists T such that $(\nu\tilde{c})(Q!m_1.F_1 \dots !m_n.F_n) \xrightarrow{\tau} T$ and $S R T$;

(2) if $(\nu\tilde{b})(P!m_1.E_1 \dots !m_n.E_n) R (\nu\tilde{c})(Q!m_1.F_1 \dots !m_n.F_n)$ and $(\nu\tilde{b})(P!m_1.E_1 \dots !m_n.E_n) \xrightarrow{\alpha} S$, then there exists T such that $(\nu\tilde{c})(Q!m_1.F_1 \dots !m_n.F_n) \xrightarrow{\alpha} T$, where α is not a higher order action, $\alpha \neq \tau$ and $S R T$;

(3) if $(\nu\tilde{b})(P!m_1.E_1 \dots !m_n.E_n) R (\nu\tilde{c})(Q!m_1.F_1 \dots !m_n.F_n)$ and $(\nu\tilde{b})(P!m_1.E_1 \dots !m_n.E_n) \xrightarrow{a(\bar{m}.0)} S$, then there exists T such that $(\nu\tilde{c})(Q!m_1.F_1 \dots !m_n.F_n) \xrightarrow{a(\bar{m}.0)} T$ and $S R T$, where m is a fresh name;

(4) if $(\tilde{v}b)(P|m_1.E_1|\dots|m_n.E_n) R (\tilde{v}c)(Q|m_1.F_1|\dots|m_n.F_n)$ and $(\tilde{v}b)(P|m_1.E_1|\dots|m_n.E_n) \xrightarrow{(\tilde{v}e)\tilde{a}(E)} S$, then there exist T, F, \tilde{f} such that $(\tilde{v}c)(Q|m_1.F_1|\dots|m_n.F_n) \xrightarrow{(\tilde{v}f)\tilde{a}(F)} T$ and $(\tilde{v}e)(S|m.E) R (\tilde{v}f)(T|m.F)$, where m is a fresh name.

Case (1): We have $(\tilde{v}b)(P|m_1.E_1|\dots|m_n.E_n) \xrightarrow{e} (\tilde{v}b)(P'|m_1.E_1|\dots|m_n.E_n) \equiv S$, hence $P \xrightarrow{e} P'$. Since $(\tilde{v}b)(P|m_1.E_1|\dots|m_n.E_n) R (\tilde{v}c)(Q|m_1.F_1|\dots|m_n.F_n)$, $(\tilde{v}b)(P|\Pi_{k_1}m_1.E_1|\dots|\Pi_{k_n}m_n.E_n) \approx_{nor} (\tilde{v}c)(Q|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n)$ for any $k_1, \dots, k_n \in N$, we have $(\tilde{v}b)(P|\Pi_{k_1}m_1.E_1|\dots|\Pi_{k_n}m_n.E_n) \xrightarrow{e} (\tilde{v}b)(P'|\Pi_{k_1}m_1.E_1|\dots|\Pi_{k_n}m_n.E_n) \equiv S'$, $(\tilde{v}c)(Q|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n) \xrightarrow{e} (\tilde{v}c)(Q'|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n) \equiv T'$ and $S' \approx_{nor} T'$. Therefore $(\tilde{v}c)(Q|m_1.F_1|\dots|m_n.F_n) \xrightarrow{e} (\tilde{v}c)(Q'|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n) \equiv T$ and $S R T$.

Case (2): There are two subcases: (a) $\alpha \notin \{m_1, \dots, m_n\}$; (b) $\alpha \in \{m_1, \dots, m_n\}$.

Subcase (a): Similar to Case (1).

Subcase (b): Suppose $\alpha = m_1$. Since m_1, \dots, m_n are fresh names, we have $(\tilde{v}b)(P|m_1.E_1|\dots|m_n.E_n) \xrightarrow{m_1} (\tilde{v}b)(P|E_1|m_1.E_1|\dots|m_n.E_n) \equiv S$. Since $(\tilde{v}b)(P|m_1.E_1|\dots|m_n.E_n) R (\tilde{v}c)(Q|m_1.F_1|\dots|m_n.F_n)$, $(\tilde{v}b)(P|\Pi_{k_1+1}m_1.E_1|\dots|\Pi_{k_n}m_n.E_n) \approx_{nor} (\tilde{v}c)(Q|\Pi_{k_1+1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n)$ for any $k_1, \dots, k_n \in N$, we have $(\tilde{v}b)(P|\Pi_{k_1+1}m_1.E_1|\dots|\Pi_{k_n}m_n.E_n) \xrightarrow{m_1} (\tilde{v}b)(P|E_1|\Pi_{k_1}m_1.E_1|\dots|\Pi_{k_n}m_n.E_n) \equiv S'$, $(\tilde{v}c)(Q|\Pi_{k_1+1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n) \xrightarrow{m_1} (\tilde{v}c)(Q|F_1|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n) \equiv T'$ and $S' \approx_{nor} T'$. Therefore $(\tilde{v}c)(Q|m_1.F_1|\dots|m_n.F_n) \xrightarrow{m_1} (\tilde{v}c)(Q|F_1|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n) \equiv T$ and $S R T$.

Case (3): We have $(\tilde{v}b)(P|m_1.E_1|\dots|m_n.E_n) \xrightarrow{a(\tilde{m}.0)} (\tilde{v}b)(P'|m_1.E_1|\dots|m_n.E_n) \equiv S$, where m is a fresh name, hence $P \xrightarrow{a(\tilde{m}.0)} P'$. Since $(\tilde{v}b)(P|m_1.E_1|\dots|m_n.E_n) R (\tilde{v}c)(Q|m_1.F_1|\dots|m_n.F_n)$, $(\tilde{v}b)(P|\Pi_{k_1}m_1.E_1|\dots|\Pi_{k_n}m_n.E_n) \approx_{nor} (\tilde{v}c)(Q|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n)$ for any $k_1, \dots, k_n \in N$, we have $(\tilde{v}b)(P|\Pi_{k_1}m_1.E_1|\dots|\Pi_{k_n}m_n.E_n) \xrightarrow{a(\tilde{m}.0)} (\tilde{v}b)(P'|\Pi_{k_1}m_1.E_1|\dots|\Pi_{k_n}m_n.E_n) \equiv S'$, $(\tilde{v}c)(Q|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n) \xrightarrow{a(\tilde{m}.0)} (\tilde{v}c)(Q'|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n) \equiv T'$ and $S' \approx_{nor} T'$. Therefore $(\tilde{v}c)(Q|m_1.F_1|\dots|m_n.F_n) \xrightarrow{a(\tilde{m}.0)} (\tilde{v}c)(Q'|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n) \equiv T$ and $S R T$.

Case (4): We have $(\tilde{v}b)(P|m_1.E_1|\dots|m_n.E_n) \xrightarrow{(\tilde{v}e)\tilde{a}(E)} (\tilde{v}b)(P'|m_1.E_1|\dots|m_n.E_n) \equiv S$, hence $P \xrightarrow{(\tilde{v}e)\tilde{a}(E)} P'$. Since $(\tilde{v}b)(P|m_1.E_1|\dots|m_n.E_n) R (\tilde{v}c)(Q|m_1.F_1|\dots|m_n.F_n)$, $(\tilde{v}b)(P|\Pi_{k_1}m_1.E_1|\dots|\Pi_{k_n}m_n.E_n) \approx_{nor} (\tilde{v}c)(Q|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n)$ for any $k_1, \dots, k_n \in N$, we have $(\tilde{v}b)(P|\Pi_{k_1}m_1.E_1|\dots|\Pi_{k_n}m_n.E_n) \xrightarrow{(\tilde{v}e)\tilde{a}(E)} (\tilde{v}b)(P'|\Pi_{k_1}m_1.E_1|\dots|\Pi_{k_n}m_n.E_n) \equiv S'$, there exist T', F, \tilde{f} such that $(\tilde{v}c)(Q|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n) \xrightarrow{(\tilde{v}f)\tilde{a}(F)} (\tilde{v}c)(Q'|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n) \equiv T'$ and $(\tilde{v}e)(S'|m.E) \approx_{nor} (\tilde{v}f)(T'|m.F)$, where m is a fresh name. Therefore $(\tilde{v}c)(Q|m_1.F_1|\dots|m_n.F_n) \xrightarrow{(\tilde{v}f)\tilde{a}(F)} (\tilde{v}c)(Q'|\Pi_{k_1}m_1.F_1|\dots|\Pi_{k_n}m_n.F_n) \equiv T$ and $(\tilde{v}e)(S|m.E) R (\tilde{v}f)(T|m.F)$. \square

References

- [1] Z. Cao, A uniform reduction equivalence for process calculi, in: Proceedings of APLAS2004, in: Lecture Notes in Computer Science, vol. 3302, Springer-Verlag, 2004, pp. 179–195.
- [2] Z. Cao, More on bisimulations for higher order π -calculus, in: Proceedings of FOSSACS2006, in: Lecture Notes in Computer Science, vol. 3921, 2006, pp. 63–78.
- [3] Z. Cao, A spatial logical characterisation of context bisimulation, in: Proceedings of ASIAN2006, in: Lecture Notes in Computer Science, vol. 4435, 2006, pp. 232–240.
- [4] W. Ferreira, M. Hennessy, A.S.A. Jeffrey, A theory of weak bisimulation for core CML, Journal of Functional Programming 8 (5) (1998) 447–491.
- [5] C. Fournet, G. Gonthier, A hierarchy of equivalences for asynchronous calculi, in: Proceedings of ICALP1998, in: Lecture Notes in Computer Science, vol. 1443, Springer-Verlag, 1998.
- [6] C. Fournet, C. Laneve, Bisimulation in the join-calculus, Theoretical Computer Science 266 (1–2) (2001) 569–603.
- [7] J.C. Godskesen, T. Hildebrandt, V. Sassone, A calculus of mobile resources, in: Proceedings of CONCUR2002, in: Lecture Notes in Computer Science, vol. 2421, Springer Verlag, 2002, pp. 272–287.
- [8] T. Hildebrandt, J.C. Godskesen, M. Bundgaard, Bisimulation congruences for homer—a calculus of higher order mobile embedded resources. Technical Report TR-2004-52, IT University of Copenhagen, 2004.
- [9] A. Jeffrey, J. Rathke, Contextual equivalence for higher-order π -calculus revisited, Logical Methods in Computer Science 1 (1:4) (2005) 1–22.
- [10] Ivan Lanese, Jorge A. Perez, Davide Sangiorgi, Alan Schmitt: On the expressiveness and decidability of higher-order process calculi, in: Proceedings of LICS2008, IEEE Computer Society, 2008, pp. 145–155.
- [11] Y. Li, X. Liu, Towards a theory of bisimulation for the higher-order process calculi, Journal of Computer Science and Technology 19 (3) (2004) 352–363.
- [12] M. Merro, M. Hennessy, A bisimulation-based semantic theory of safe ambients, ACM Transactions on Programming Languages and Systems 28 (2) (2006) 290–330.
- [13] R. Milner, A calculus of communication systems, in: Lecture Notes in Computer Science, vol. 92, Springer-Verlag, 1980.
- [14] R. Milner, J. Parrow, D. Walker, A calculus of mobile processes, (Part I and II), Information and Computation 100 (1992) 1–77.
- [15] J. Parrow, An introduction to the π -calculus, in: J. Bergstra, A. Ponse, S. Smolka (Eds.), Handbook of Process Algebra, North-Holland, Amsterdam, 2001.
- [16] D. Sangiorgi, Expressing mobility in process algebras: first-order and higher-order paradigms, Ph.D thesis, University of Edinburgh, 1992.
- [17] D. Sangiorgi, Bisimulation in higher-order calculi, Information and Computation 131 (2) (1996) 141–178.
- [18] D. Sangiorgi, N. Kobayashi, E. Sumii, Environment bisimulations for higher-order languages, in: Proceedings of LICS2007, IEEE Computer Society, 2007, pp. 293–302.
- [19] A. Schmitt, J. Stefani, The Kell calculus: a family of higher order distributed process calculi, in: Proceedings of GC2004, in: Lecture Notes in Computer Science, vol. 3267, Springer-Verlag, 2004, pp. 146–178.